IN-62
179493
P. 76

NASA Technical Memorandum 109003

A PC-Based Simulation of the National Transonic Facility's Safety Microprocessor

J. J. Thibodeaux
NASA Langley Research Center
Hampton, Virginia

W. A. Kilgore and S. Balakrishna
ViGYAN, Inc.
Hampton, Virginia

JULY 1993

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

SUMMARY

A short study was undertaken to demonstrate the feasibility of using a state-of-the-art off-the-shelf high speed personal computer (PC) for simulating a microprocessor presently used for windtunnel safety purposes at Langley Research Center's National Transonic Facility (NTF). Currently, there is no active display of tunnel alarm/alert safety information provided to the tunnel operators, but rather such information is periodically recorded on a process monitoring computer printout. While this recording does provide adequate data, generally, it does not provide on-line centralized situational information nor permit rapid identification of safety operational violations which are able to halt tunnel operations. It was therefore decided to simulate the existing safety control algorithms and briefly evaluate a real-time display which could provide both current equipment position as well as trouble shooting information. The initial step of this study was to transform the existing microprocessor software into QuickBasic (version 4.5) software for programming into the selected personal computer. After this, hardware checkout was begun. For demonstration, and final evaluation, an already existing PC-based signal input generator was connected to the simulated safety microprocessor. This device was used to simulate inputs that would normally be supplied by tunnel transducers. This study together with another recent control application has shown these inexpensive computers to be reliable, and can be used for sophisticated tasks

such as information display and protection of expensive wind tunnel equipment at very modest costs. Furthermore, it was learned that safety code transformation along with generating a highly accurate complex situational/warning display could be achieved without a great deal of effort. Included in this short document is the PC source code for the simulation, and a selected real-time alert display along with a brief explanation of the various tunnel safety control checks performed by the NTF's safety microprocessor.

INTRODUCTION

The National Transonic Facility, which is the world's largest cryogenic wind tunnel, was commissioned in May 1982. References 1 and 2 describe the operational characteristics of this facility along with some of its special features. Until recently, this elaborate research test facility was still using the microprocessor systems that were designed using 1970's type of computer technology. Five of these microprocessors that were used to control and regulate Mach number, temperature, pressure, fan speed, and test section configuration have been recently replaced with 1990's computing technology. The many recent technological advances and innovations that have taken place in computers has helped initiate the changeover to this newer equipment. A noteworthy amount of these advances have also occurred in personal computers (PC's). Among these is the capability for high frequency computation at greater signal bit resolution while simultaneously realizing decreasing costs and significantly increased reliability. Currently, a PC clone (12 Mhz CPU, EGA video, with a hard disk, and an eight channel digital-to-analog converter) computer has been operating Langley Research Center's 0.3-meter cryogenic wind tunnel since 1988 (reference 3). This system has now been operating very efficiently for several thousand hours without any major down time. Rarely has board level replacement been necessary, but when it was once required, quick low cost repairs resulted.

To address the need for a real-time display of NTF safety operational information, and to better understand the existing safety microprocessor code for future enhancement purposes, it was decided to research the application of an advanced high-speed personal computer for use as a potential replacement for the existing NTF tunnel safety microprocessor. The system consisted of a high speed 386-chip personal computer, 16-bit analog-to-digital, and digital input/output hardware. A second PC was used as a signal generator (reference 4) to provide inputs for software checkout, and display evaluation of the simulated safety microprocessor. Results as well as the source code listing of those operational tests are presented here.

SYMBOLS

ALPHAB	Bottom test section wall position, degrees
ALPHAT	Top test section wall position, degrees
BETABF	Bottom far-side model support wall position, degrees
BETABN	Bottom near-side model support wall position, degrees
BETATF	Top far-side model support wall position, degrees
BETATN	Top near-side model support wall position, degrees
CD	Radial distance between model support wall pivot point and corresponding reentry flap tip, inches
DISABE	Pitch limit disable logic switch
GAMABF	Bottom far-side reentry flap position, degrees
GAMABN	Bottom near-side reentry flap position, degrees
GAMAFS	Far-side vertical wall reentry flap position, degrees
GAMANS	Near-side vertical wall reentry flap position, degrees
GAMATF	Top far-side reentry flap position, degrees
GAMATN	Top near-side reentry flap position, degrees
MACHSQ	Mach number squared
PRATO	Ratio of total to static pressure (XPRES/PSTAT)
PSTAT	Static pressure, psi
QCOMP	Computed tunnel dynamic pressure, psi
RHO	Density, lbm/cubic inch
ROL	Arc sector roll angle, degrees
XGVA	Inlet guide vane angle, degrees
XNBF	Angle between bottom far-side model support wall and its pivot-to-reentry flap tip radius, degrees

XNBN	Angle between bottom near-side model support wall and its pivot-to-reentry flap tip radius, degrees
XNTF	Angle between top far-side model support wall and its pivot-to-reentry flap tip radius, degrees
XNTN	Angle between top near-side model support wall and its pivot-to-reentry flap tip radius, degrees
XPRES	Total pressure, psi
XPSP	Pitch hydraulic pump swash plate position, percent
XQDIAL	Tunnel dynamic pressure thumbwheel limit, psi
XSTRDM	Pitch angle thumbwheel negative limit, degrees
XSTRDP	Pitch angle thumbwheel positive limit, degrees
XSTRUT	Current arc sector pitch angle, degrees
(α)	Test section wall angle, degrees
(α_1)	Arc sector pitch angle, degrees
(β)	Model support wall angle, degrees
(η)	Angle between the model support wall and its pivot-to-reentry flap tip radius, degrees
(γ)	Angle between the model support wall and the reentry flap, degrees
Δ	Increment

DISCUSSION

General Comments

Until recently, microprocessor computing capability used at the NTF was approximately twenty years behind the current state-of -the-art computer industry. Devices that were used for control algorithm calculations were based on 1970's microprocessor technology which consisted of compiled software code with floating point hardware executing at a speed of approximately 100 milliseconds per control loop. All executable code resided in non-volatile read-only EPROM-based memory. Eight-bit analog-to-digital converters, which were the standard for that time period, have been recently replaced with 16-bit devices to increase input resolution. Process feedback generation was created by reading analog signals by way of analog-to-digital converters while display information was created by digital output devices. For operational purposes, a clocked/watch dog timer interrupt structure was used in this earlier microprocessor system. Control algorithm execution was periodic at a fixed clock frequency which was checked every period to ensure correct cycle time for self-diagnostic purposes. This execution time was slowed somewhat due to continuous memory sharing by an internal communication controller module. Memory accesses to/from these locations were interleaved between the communication controller and the main microprocessor controller.

Because of hardware and software difficulties with these older microprocessors as well as recent decreases in costs and technical advances in computing technology, it was necessary to replace this older equipment. While much of the old hardware architecture has changed, the software has remained the same. A research activity was initiated to investigate the potential of simulating the NTF's safety microprocessor algorithms in a present day low-cost high speed (386 micro-chip) off-the-shelf personal computer, and explore the capability of such a device for PC-based safety control and display. This effort was also directed at assessing the existing software and eventually modifying selected areas of code to more nearly match the tunnel testing envelope. It was especially desirable to generate and explore the benefits of a real-time display of tunnel safety operational information.

Figure 1 is a schematic of equipment used for this study while figure 2 is a photograph of this same hardware. The hardware-in-the-loop simulator, itself PC based, was used to generate signals introduced to the safety micro simulation via the analog wiring box. These input signals were subsequently used to drive the various test alarms/alerts which will be discussed in greater detail later. The base addresses for the three a-to-d boards (DT-2801/5716A) as well as the base addresses for the two DIO boards (DT-2817) of the safety micro are shown in figure 1. Each of these DIO boards have four port registers (ref. 5). For the study, one DIO board was configured for accepting inputs while the other was setup to send alarm discretes which simulate relay signals to the programmable ladder-logic sequencers. There are three such sequencers currently used at the NTF. Sequencers are used to prevent inadvertent damage to equipment if improper operational sequences are attempted. The first DIO board (base address H228) was configured such that two ports were dedicated for accepting simulated pitch thumbwheel limits (plus and minus) while the other two ports were used additively to generate the tunnel dynamic pressure alarm limit. The second board (base address H250) was configured strictly for handling relay discrete outputs intended for sequencer specific alarm action. The four discrete alarms selected for output were as follows:

1) Tunnel dynamic pressure exceeding the thumbwheel limit (+17.6 psi).
2) Arc sector pitch angle exceeding the plus (+16.0 deg.) and minus (-8.0 deg.) thumbwheel limits.
3) Model support walls bottom nearside/farside not in synchronous operation (absolute difference in near and far angles greater than 0.24 deg.).
4) Test section wall bottom/model support wall bottom nearside/reentry flap bottom nearside collision (see figure 3).

The number of alarm outputs thus generated was restricted to four due to a limited number of slots available in the selected computer. These alarms were selected as representative of those used in the NTF. Any of the other conditions could have been chosen for output. If this microprocessor were to be actively used for tunnel safety monitoring, it would be necessary to free up an additional computer slot and include another DIO board so as to accommodate other alarm outputs. All safety alerts of the program however are visually presented on the display screen.

The initial phase of this study consisted of coding all safety micro algorithms in the desk top PC. The QuickBasic code (ref. 6) thus developed paralleled the general format of the existing safety microprocessor Fortran 77 software. A basic difference between the two sets of codes was that the PC software executed in the same manner as the old micro software but without using subroutine calls. In effect, the PC code is an in-line layout of the original microprocessor software allowing efficient updating and development. Appendix A contains the commented simulator source code listing.

PC software coding began with the input of known conversion constants and values of equipment physical limits, cosine's of angles as well as analog-to-digital and digital input/output device base addresses. After the input of constants, the analog-to-digital boards were set up for reading the analog signals from simulated field transducers generated by the input simulator. A total of nineteen inputs were thus brought into these converters. A status check on the registers of the a-to-d boards as well as the digital I/O boards was performed to ascertain if a fatal board error had occurred. Because of the total number of signals, the use of three a-to-d boards (DT-2801/5716A), and two DIO boards (DT-2817) was required to complete this simulation. It was these feedback signals and calculations using these inputs which were tested to ensure that no equipment violations occurred while operating the tunnel. Signals thus introduced were converted into real engineering numbers by directing them through calibration equations, and these current values were then printed to the screen for display. Figure 4 is the display (in an unalarmed state) used for demonstration purposes in this simulation. The maximum and minimum operational ranges for the particular equipment or variable are shown to the immediate right. These values were overwritten in bright red in the event of an alarm situation. The current value of the variable in question is displayed immediately to the left within the parentheses. Various alerts corresponding to the different software/hardware tests are shown later as examples. At the bottom right are four zero digits. Each individual digit cycles between 0 and 1 corresponding to the particular digital port selected for outputting an alarm signal. From left to right, the digits correspond to "TUNNEL Q EXCEEDED", "STRUT THUMBWHEEL VALUE EXCEEDED", "MODEL SUPPORT WALL NOT IN SYNC", and "TEST SECTION WALL BOTTOM/MODEL SUPPORT WALL BOTTOM FAR/REENTRY FLAP BOTTOM FAR WILL COLLIDE" alarms.

After conversion to engineering units of the incoming data, one of the first calculations performed in the program was the determination of the radius arm "CD" of figure 3. This is commented as subroutine "CALCD" in the software listing of

Appendix A. For the displaced model support wall and reentry flap, this is the distance from the tip of the reentry flap to the pivot point of the model support wall. Four separate calculations are performed to account for the top far and nearside reentry flaps as well as the bottom near and farside flaps. Subsequent to this, the angles "XNTF, XNTN, XNBF, and XNBN" formed by the model support walls and the respective radius arms "CD" previously determined were computed. These parameters were computed in subroutine "CALN", and then used in "TEST 02" to prevent test section walls from colliding. Collision was avoided by maintaining the proper distances between the top and bottom test section walls and the model support walls as well as the distance between the test section walls and the reentry flaps. This test will be derived and discussed further in the software logic.

In order to prevent exceeding the tunnel Mach number of 1.25 and the operational tunnel dynamic pressure (1/2*RHO*V^2), two separate computations were required. The square of the tunnel operational Mach number was determined in "MACHIN" by using the ratio of total to static pressure. If this was violated, then a warning was printed to the screen. Following this, the computed dynamic pressure was determined and later compared to a set of thumbwheel input values that were brought in through the digital input ports. These inputs correspond to the desired upper limit of tunnel operational dynamic pressure. Likewise, if a violation of dynamic pressure occurred, a red colored warning was written to the screen. This will be shown later in "TEST 07" of the software.

The various tests programmed within the safety micro are referred to as TEST 01, TEST 02, TEST 03, TEST 05, TEST 06, TEST 07, TEST 10, and TEST 11. All of these tests were exercised in the simulation except TEST 05 which has been deactivated in the tunnel software. Its description is given below. As can be seen, some of the tests numbers are missing. This was a result of the evolutionary knowledge gained with tunnel operation, and experience. The following is a brief discussion of each of the various software tests.

Software Logic

TEST 01 examines the test section top and bottom wall angular positions relative to zero i.e. parallel to the tunnel centerline. Figure 3 provides a sideview of the bottom test section moveables of the NTF. The sign convention selected for the moveables was such that angular movement away from the tunnel centerline was considered positive while toward the centerline was negative. The exception to this convention was the angle (γ) which was considered to always be positive. The dashed lines represent the angular movement of these devices which are housed inside of the test section. If either of the walls was not within +-0.02 degrees of zero position, a warning was created. This test prevents model access housings from being accidentally extended into the test section unless the top and bottom walls are very near their zero position. Figure 5 shows the alarms if either the top or bottom test section walls was greater than or equal to +0.02 degrees. Similarly, figure 6 constitutes the warnings if they were less than or equal to -0.02. It should be noted that in either case an error of 0.001 degrees has caused the alert/alarm conditions to occur. This gives an idea of the accuracy of this simulation.

TEST 02 is a two-part test used to prevent the test section walls, and model support walls, or reentry flaps from colliding. There were two separate calculations within this main test. These equations are developed here for future reference. Figure 3, as mentioned earlier, provides the sideview, dimensions, and angular travel of the test section movables.

The first part of this test calculates the clearance of the downstream end of the test section wall and the upstream end of the model support wall. A safety interlock was initiated whenever this clearance, YB-YA (from the sideview), became less than or equal to 0.5 inches. This clearance was initially two inches, but due to knowledge gained during operations, it was reduced.

From the figure, the test section wall angle (α) can be either positive or negative while the model support wall angle (β) is defined to be negative while traveling toward the stream flow. Likewise, it is clear that:

$$YA = a + 300 \sin(\alpha)$$

and:

$$YB = b + 132 \sin(\beta).$$

Therefore, substituting these into the safety criteria we get:

$$b + 132 \sin(\beta) - (a + 300 \sin(\alpha)) <= 0.5$$

or

$$b - a + 132 \sin(\beta) - 300 \sin(\alpha) <= 0.5.$$

We see in the figure, that b - a = 9.276 inches which happens to be the distance between the pivot points of the test section wall and the model support wall. Since the angles (α), and (β) will be small, sin (β) = (β) (in radians), and sin (α) = (α) (in radians). Consequently, the first equation of this two part test becomes:

$$9.276 + 132\,(\beta) - 300\,(\alpha) <= 0.5 \text{ in.}$$

In programming terms, this becomes:

```
TEST2(1) = KCL + KLENBD * X - (KLENA * Y).
```

The value of TEST2(1) is then compared to 0.5, and an alarm was created if this was less than or equal to 0.5.

Since there are four model support walls (top farside and nearside and bottom farside and nearside), this calculation must be performed four times in order to examine the four possible situations.

The second criteria of TEST 02 ensures that the leading edges of the four reentry flaps do not get any closer than 0.5 inches to the downstream end of the upper and lower test section walls. As before, this criteria was reduced to 0.5 inches from two inches due to numerous alarms experienced during shakedown. This portion of the test was seen to be vastly more complex trigonometrically than the first part due to the changing lateral distance described by the reentry flaps and model support walls during movement.

Mathematically, in terms of the symbols of figure 3, the alarm condition can be expressed as:

$$YA1 - YC <= 0.5 \text{ in.}$$

If, at anytime during movement, the left side of the equation becomes less than or equal to 0.5 inches, the microprocessor must alarm by setting the appropriate relay logic in the sequencer. Again from figure 3, it can be seen that:

$$\sin(\beta - (+\eta)) = YC/CD$$

or $$\sin(-1(-\beta + \eta)) = YC/CD.$$

Since sin (-angle) = -sin (angle), we can write:

$$-\sin(-\beta + \eta) = YC/CD.$$

For small angles, $\sin(-\beta + \eta) = (-\beta + \eta)$, and therefore

$$YC = -CD\,(-\beta + \eta).$$

The above is true for small angles expressed in radians. By definition (β), was restricted to having negative values of 0 to -4.5 degrees. Because the voltage from the model support wall sensor in the tunnel gives a positive voltage, a negative sign is afixed by way of calibration within the microprocessor.

From the above, CD and (η) are the only two unknowns, but they were determined as shown below.

By the law of cosines for the triangle CBD in figure 3, we see that:

$$CD^2 = BD^2 + CB^2 - 2(BD)(CB)\cos(180 - \gamma).$$

Substituting the constants $BD^2 = 132^2 = 17424$, $CB^2 = 60^2 = 3600$, and solving for CD we get:

$$CD = (17424 + 3600 - 2(132)(60)\cos(180 - \gamma))^{1/2}$$

or $$CD = (21024 - 15840\cos(180 - \gamma))^{1/2}.$$

Since $\cos(180 - \gamma) = -\cos(\gamma)$, the above is expressed as:

$$CD = (21024 + 15840\cos(\gamma))^{1/2}.$$

In program variable form, this becomes:

```
CDX = SQRT (KCD1 + KCD2 * INTCOS)
```

where INTCOS = cos (γ).

In order to find (η), we used the law of sines which gives:

$$CD/\sin(180 - \gamma) = CB/\sin(\eta).$$

Since CD was determined from above and CB = 60 in.,

$$(\eta) = \sin^{-1}(60/CD \sin(180 - \gamma))$$

or

$$(\eta) = \sin^{-1}(60/CD \sin(\gamma))$$

because $\sin(180 - \gamma) = \sin(\gamma)$. For the top nearside model support wall, this was written in the program as:

$$XNTF = (\eta) * KRAD$$

where KRAD = 0.017432 radians per degree.

The angles (α) and (β) (always negative by definition) are known and since we can solve for CD and (η) from the above equations, the only unknown is L. This can be calculated by again studying figure 3. It is obvious that the distance X1 will change as a function of the reentry flap and model support wall positions and consequently so will L.

It remains now only to derive an expression that will take into account this variation. By looking at the figure, it can readily be seen that:

$$\tan(\alpha) = L/(432 - X1)$$

where $X1 = CD \cos(-\beta - (+\eta)) = CD \cos(-1(\beta + \eta))$. Since the $\cos(-\text{angle}) = \cos(\text{angle})$, $X1 = CD \cos(\beta + \eta)$.

Therefore:

$$L = (432 - CD \cos(\beta + \eta)) \tan(\alpha).$$

For small (α), the $\tan(\alpha) = (\alpha)$ (in radians) and the $\cos(\beta + \eta) \cong 1$ since $(\beta + \eta)$ is a very small angle. Consequently:

$$L = (432 - CD)\ (\alpha)\ \text{(in radians)}.$$

Once again from the figure:

$$YA1 = 9.276 - L.$$

After substituting of L, the above becomes:

$$YA1 = 9.276 - (432 - CD)\ (\alpha).$$

Previously, we saw that:

$$YC = -CD\ (-\beta + \eta).$$

Inputting these last two equations into the alarm criteria i.e.

$$YA1 - YC <= 0.5,$$

we get:

$$9.276 - (432 - CD)\ (\alpha) - (-CD\ (-\beta + \eta)) <= 0.5$$

or $$9.276 + CD\ (-\beta + \eta) - (432 - CD)\ (\alpha) <= 0.5.$$

Because the sensors only provide positive voltages, the negative sign on (β) was introduced inside the microprocessor by way of the calibration curve. Thus this equation becomes:

$$9.276 + CD\ (\beta + \eta) - (432 - CD)\ (\alpha) <= 0.5\ \text{in.}$$

In programming terms, this becomes:

```
        TEST2(2) = KCL + CDTF * (X + XNTF) - (KSTA36 - CDTF) * Y
```

where X, XNTF an Y must be in radians. Since there are top and bottom test section walls as well as near and farside model support walls and reentry flaps, this computation was also performed four separate times.

Figure 7 shows the alarms that have been created to warn of impending collision of the top and bottom test section walls with their respective model support walls, and reentry flaps. At the bottom right of this figure is shown the digital value of 0 0 0 1.

The value of one in this position corresponds to setting true of the alarm logic for "TSWB/MSWBF/REFBF COLLIDE" previously mentioned.

In order to ensure that the model support walls are always within 0.24 degrees of one another, TEST 03 constantly computes the absolute angular difference between the bottom nearside and farside model support walls. Additionally, this same absolute difference is compared to 0.24 degrees for the top nearside and farside model support walls. In event that this limit was exceeded, a red warning message ("MSW NOT IN SYNC") was printed to the screen. This warning is commonly shared by both bottom and top model support walls. Figures 8, and 9 show this warning displayed at the very bottom of the screen. A comparison of the model support walls present values shows how closely the alarm occurs after exceeding the 0.24 difference. Additionally, since the third digit in figure 9 is unity, this port was outputting to the sequencer for alarm action for the bottom walls. Because of a shortage of output ports, this could not be done for the top model support walls.

TEST 05 is a simple software check of the two independent side wall reentry flap positions. The code routinely checks to ensure that the angles of the side flaps were not less than or equal to zero degrees. This precludes near and farside flap extension into the flow stream. Initially, these side flaps were intended to be used with slotted side walls of the test section. These slotted walls have not been installed at the NTF, and consequently the side flap actuators are locked in a safe position and are not normally moved during aerodynamic testing. This test remains in software, however, for future use. Figure 10 gives the two simulation warnings obtained in the event of a position violation.

The fundamental objective of TEST 06 was to prevent any striking or interference between the model arc sector strut and the top or bottom tunnel model support walls of the test section. The complexity of this test is best understood by a brief derivation of these two similar equations. Before undertaking this task, an understanding of how safety margins for the arc sector, and the model support walls were determined is necessary. It was known that the arc sector hydraulic shuttle actuator would require 0.125 seconds to achieve the failsafe position for stopping the arc sector, and 0.33 seconds would be required for the ladder-logic sequencer to respond to an interference condition. A 10 percent margin was added to the sum of these times which yields approximately 0.5 seconds for halting the interference situation. Since the arc sector was assumed to travel at 4 degrees per second, 2 degrees (4 deg/sec X 0.5 sec) of pitch margin was desired. Likewise, a value of 0.15 degrees (0.3 deg/sec X 0.5 sec) was used for the model support wall margin of safety. Reference 7 was used to obtain a better understanding of the operation of this part of the program. A plot of model support wall travel versus the arc sector angular motion (figure 11) along with the designed margins of safety (dashed-lines) is provided here for clarity. The 0.15 degree MSW offset dictated that the top and bottom MSW/Arc sector safety lines must be parallel to the inteference lines, and by simply offsetting by this amount the desired margin of safety was achieved. Additionally, inorder to achieve the maximum pitch operating range, and larger safety margins for faster movements, the 2 degree pitch safety margin was included in the arc sector pump stroker position which is proportional to arc sector velocity. Therefore, the top and bottom safety margin equations were derived to be functions of these two offsets as is shown below.

From figure 11, these are linear equations through the offset points (X1,Y1), and (X2,Y2) which can be expressed as:

$$(Y - Y1)/(Y2 - Y1) = (X - X1)/(X2 - X1)$$

where for the top:

$$X1 = (-11.5 + \Delta\alpha_1) \qquad X2 = (-10.33 + \Delta\alpha_1)$$

$$Y1 = (+0.15) \qquad Y2 = (-4.5 + 0.15) = -4.35$$

and for the bottom:

$$X1 = (19 - \Delta\alpha_1) \qquad X2 = (15 - \Delta\alpha_1)$$

$$Y1 = (+0.15) \qquad Y2 = (-4.5 + 0.15) = -4.35.$$

By letting $X = (\alpha_1)$ (the current arc sector pitch angle, deg), and

$Y = (\beta)$ (the model support wall angle, deg)

and after considerable algebraic manipulation, the top MSW/Arc sector safety margin equation becomes:

$$4.5(\alpha_1 - \Delta\alpha_1) + 1.17\beta = -51.57. \qquad (1)$$

Similarly, the bottom MSW/Arc sector safety equation is:

$$4.5(\alpha_1 + \Delta\alpha_1) \quad -4\beta = 84.9 \qquad (2)$$

The constants shown above (-51.57 and 84.9) are the values that the computed left side of equations (1), and (2) were tested against in

TEST 06. If at anytime, the left side of equation (1) was less than or equal to -51.57 or the left side of equation (2) was greater than or equal to 84.9, then a strut-to-MSW interference warning was created for the top or bottom respectively. During real tunnel operations, test section movement would be halted until the situation is corrected. Figure 12 graphically shows the physical relationship of the test section hardware to the established limits.

While the implementation of this simulation was progressing, a C of F effort was begun at the NTF. Among the numerous improvements for increased tunnel production was the replacement of the existing arc sector hydraulic system with a new variable volume high pressure power pack. Because of this, the swash plate signal of the old system had to be eliminated. Consequently, a suitable signal had to be found because TEST 06 software uses the swash plate signal for its checks. While trying to commission this new system, data from the spool position was recorded and analyzed. This signal was found to be free of noise and would require a minimum of software change in the safety microprocessor so as to continue to prevent interference between the arc sector and the top or bottom model support walls. The signal is currently used at the facility, but has not been incorporated in this simulation until more on-line experience has been acquired. Figures 13, and 14 show the red colored alarms for this condition.

TEST 07 consists of a number of simple tests that are particularly interesting to safe tunnel operation. The first of these determines whether current arc sector pitch angle has exceeded preselected positive or negative pitch thumbwheel limiting values. A disable switch provides a method whereby the pitch limits can be ignored so that these limits may be changed while the tunnel is online. The software to accomplish this has been included in this simulation. In a similar manner, the calculated value of tunnel dynamic pressure (Q) was compared against a preselected thumbwheel value of dynamic pressure. Both dynamic pressure and pitch thumbwheel limits were introduced into the simulation by way of the DIO board (H228) that was configured for input only. Mach number squared was then compared to the square of the maximum allowable Mach number (1.25). The next part of this test assesses the voltage status of the strut angle, static pressure, total pressure, and static pressure range code sensors. If voltages are not maintained within acceptable upper or lower bounds, alerts are activated. Figures 15, and 16 are provided to show these alerts. Attention is now called to the first two unity digits at the lower right of the figure 15. Both "TUNNEL Q EXCEEDED" and "STRT THM'WL XEE'ED" have been made to activate thereby setting the corresponding DIO board (H250) output ports to one. Figure 16 shows the static pressure sensor voltage range check on the lower end. A similar check was performed on the upper end of the sensor, but was not activated nor shown here. Figures 17, and 18 show the upper and lower checks of the static pressure range switching code that was performed. The switching code must remain within 3.00 and 4.50 volts so as not to disrupt operations.

The numerical values just to the left of the alarm messages give an indication of the values creating the alert activation. In either case, a difference of 0.01 volts has resulted in tripping of the alarm.

There are two parts to TEST 10. Its first part is to calculate the difference between the upper far side test section model support wall and the corresponding reentry flap, and compare that difference to 15.0 degrees. If the difference is greater than 15.0 degrees, then an alarm was directed to the sequencer which halts operation. This is repeated for the lower near, and far side reentry flaps as well. Figure 19 shows the reentry flap values to be set to 15.003, 15.008, 15.006, and 15.009. The small differenc-es (model support wall angle minus the reentry flap angle) of 0.003, 0.007, 0.005, and 0.009 degrees are causing the alerts of this condition. The second part of this short test examined the two upper and two lower reentry flap positions. If the specified reentry flap position was less than +0.1 degrees, then it was going into the tunnel slipstream, and was alarmed. Figure 20 is provided to show the occurrence of these alerts for farside and nearside top and bottom reentry flaps. The current values of 0.098 were deliberately set to demonstrate the small difference (0.001) needed for alarm activation.

The final test (TEST 11) of this program scans the model roll angle transducer and compared that to ensure that the preselected roll thumbwheel maximum, and minimum limits (+275 or -95 degrees) were not exceeded. Inside the tunnel, input sensor data for this test is derived from a highly accurate inductive type roll resolver. Since this test has not been functional during windtunnel operations, it was not activated or exercised in this simulation. The software, however, has been programmed, and can be immediately implemented if desired.

System Evaluation

To evaluate the utility of the system, several tunnel operators were given the opportunity to operate, analyze and comment on the overall effectiveness, usefulness, and potential of this system. Generally speaking, all operators responded in the affirmative that during tunnel operations such a real-time on-line device would be of benefit especially during the trouble shooting mode of alarmed conditions. Furthermore, it was indicated that because of past difficulties in positively identifying safety operational culprits, latching software should be developed as part of this effort. At a minimum, this code should provide day/time information as well as hold the troublesome condition on screen thereby allowing correlation with other corroborating data ensuring rapid correction. A hardcopy printout of this should be generated as well. This information should not be deleted from the screen until operators have properly identified, and are convinced of the validity of the alert. Only then would resetting of the computer/display be allowed.

In the event that this system is adapted at the NTF, these suggestions would be given further consideration and certainly require evaluation during tunnel operations.

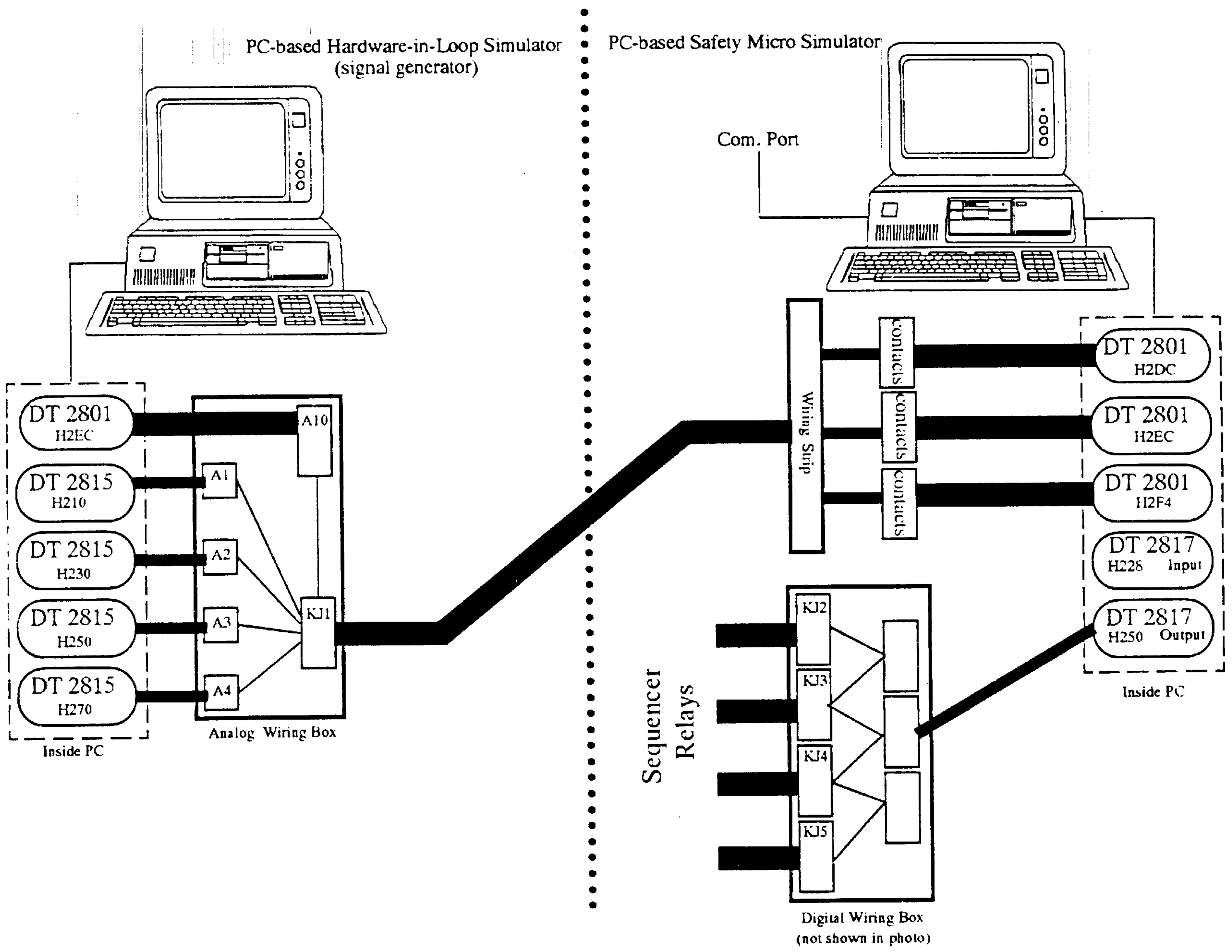
CONCLUDING REMARKS

A computer simulation of the safety microprocessor at Langley Research Center's National Transonic Facility was developed and verified. The objective of this brief study was to demonstrate the utilization of a low-cost state-of-the-art high-speed personal computer combined with digital input/output hardware for computation of complex tunnel safety algorithms. Software code comprising this system was checked by employing a setpoint signal generator which itself was a PC-based device capable of driving the simulator through its various safety tests. Each alarm/alert check residing within the code was systematically exercised and verified. A real-time display was also created which provided operational alarms, and alerts along with current tunnel equipment positional information. As per evaluation by several NTF operators, this display has potential for providing rapid on-line trouble shooting and warning capability providing that alarm correlation time is made available on the display. Additionally, operators indicated that the computer/display must not be reset until the source of the alert could be positively identified otherwise utility would be lost. This would necessarily require a latching routine that would hold the alarmed condition for trouble shooting purposes. Throughout this task, the PC was found to require relatively little hardware and programming experience for setup. Additionally, this 386-based computer system was found to be highly reliable, very accurate, and capable of extremely high frequency operation (about 50 milliseconds per program cycle).

REFERENCES

1. Bruce, W. E.: The U. S. National Transonic Facility, Parts I and II. Special Course on Cryogenic Technology for Wind Tunnel Testing, AGARD Report No. 722-R, Jul. 1985.
2. Balakrishna, S., Kilgore, W. A., and Thibodeaux, J. J.: Control of Large Cryogenic Tunnels, 17th Aerospace Ground Testing Conference, AIAA paper no. 92-3930, Jul. 1992.
3. Vigyan Inc.: Kilgore, W. A., and Balakrishna, S.: The NASA Langley Research Center 0.3-Meter Transonic Cryogenic Tunnel Microcomputer Controller Source Code, NASA CR-189556, Dec. 1991.
4. Data Translation Inc.: User Manual for DT-2801 Series, Analog and Digital I/O Boards for the IBM PC/XT/AT and Compatibles, Eleventh Edition, Sep. 1991.
5. Microsoft Corporation: Microsoft QuickBasic (version 4.5)--Programming in Basic, Document No. 410700014-450-R01-0988, Copyright 1987, 1988.
6. Sverdrup ARO, Inc.: Safety Analysis of the National Transonic Facility Integrated Operations--Microcomputer Software Safety Report, Jul. 1981.

Figure 1. PC-based Safety Microprocessor Simulation.



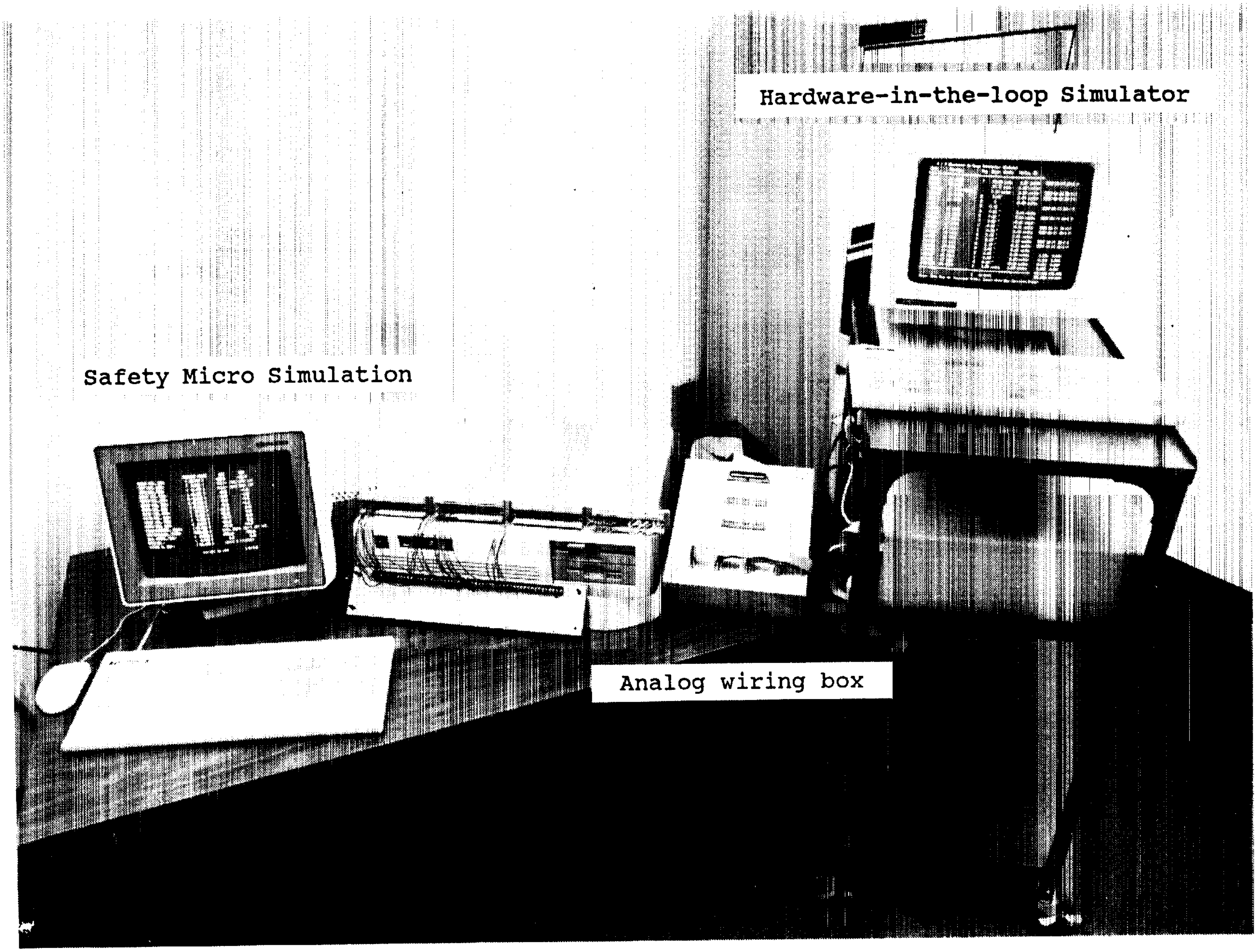
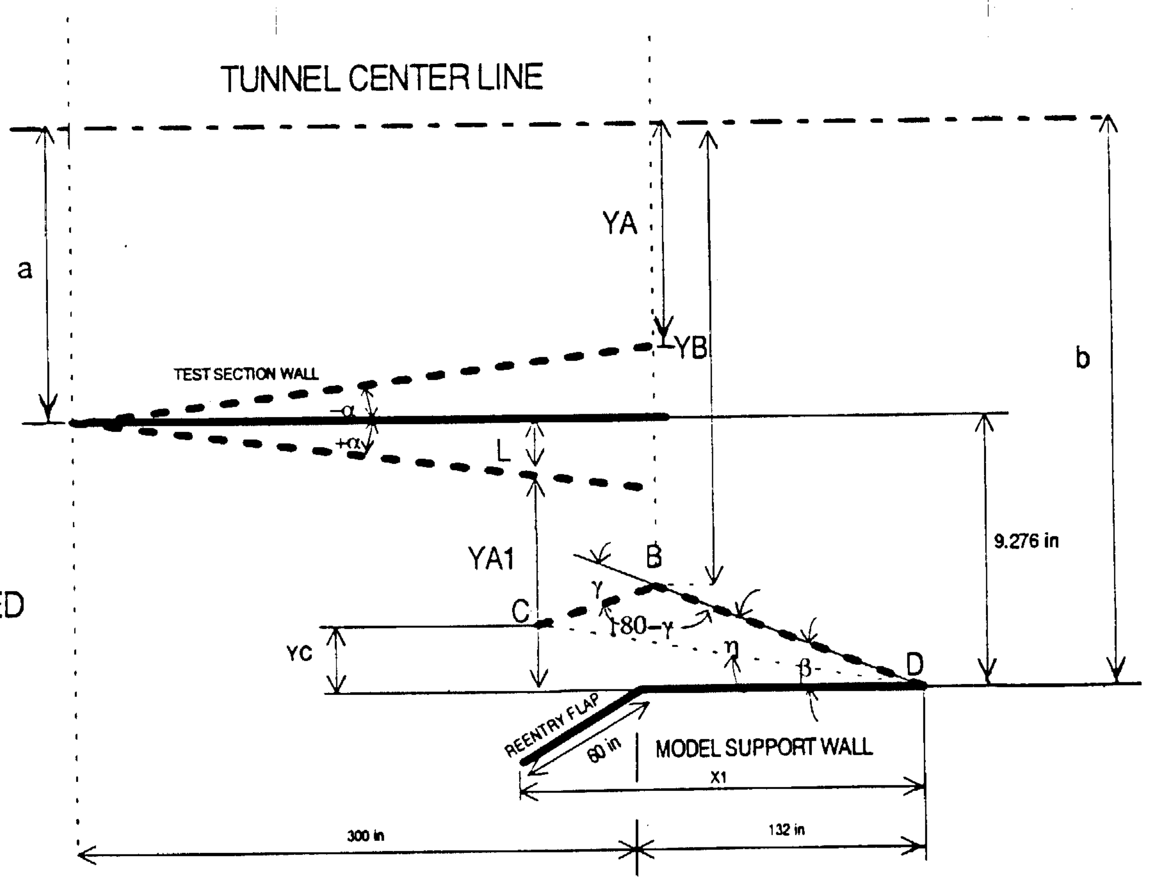


Figure 2. - PC-based safety simulation.

Figure 3. - Side view of the test section (bottom).



$-0.5 <= \alpha <= 1.0$

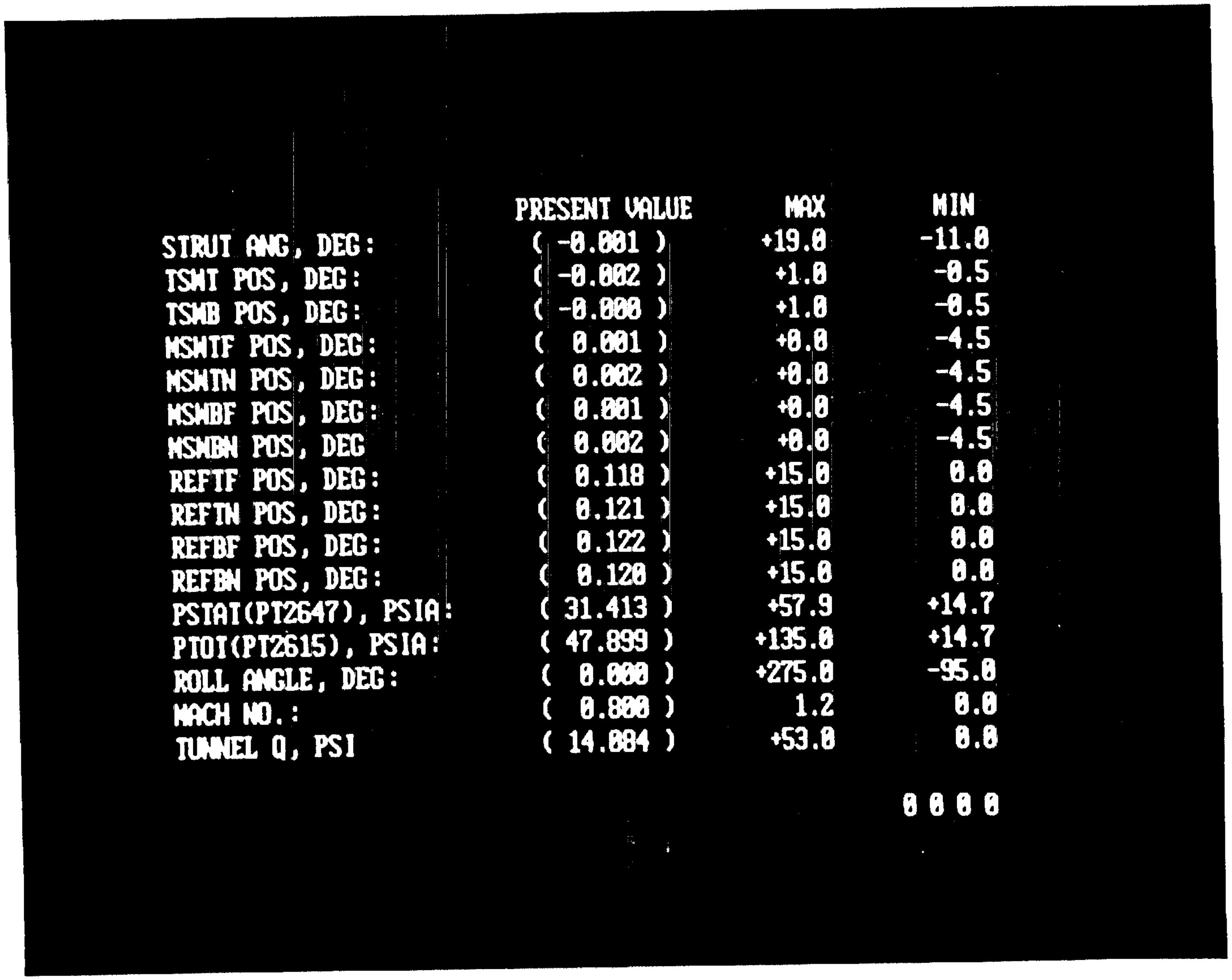
0 to -4.5 for β

0 to +15 for γ

η is always positive

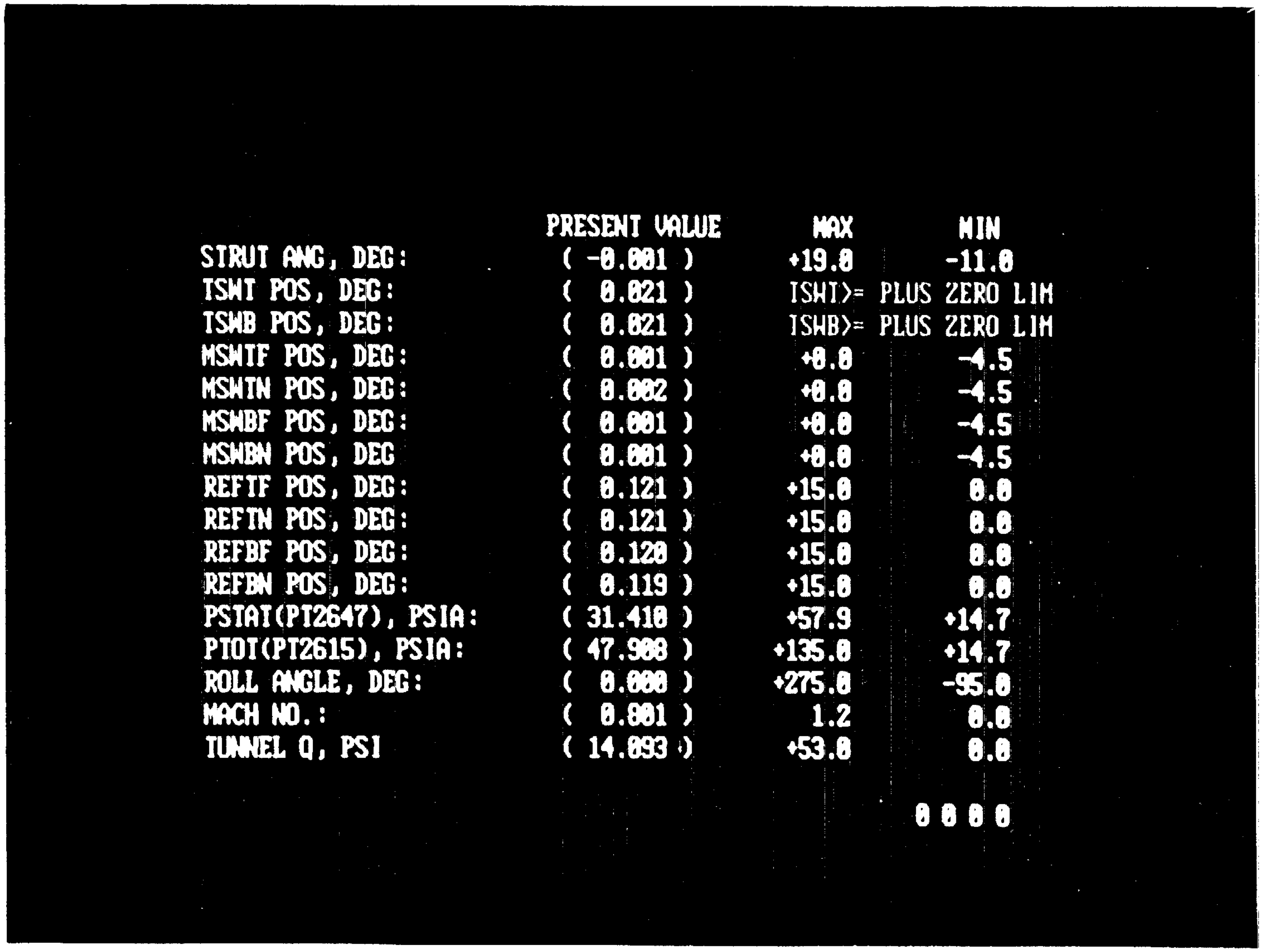
ANGULAR SIGN CONVENTION:
AWAY FROM STREAM IS POSITIVE

CB= REENTRY FLAP DISPLACED
BD=MODEL SUPPORT WALL DISPLACED



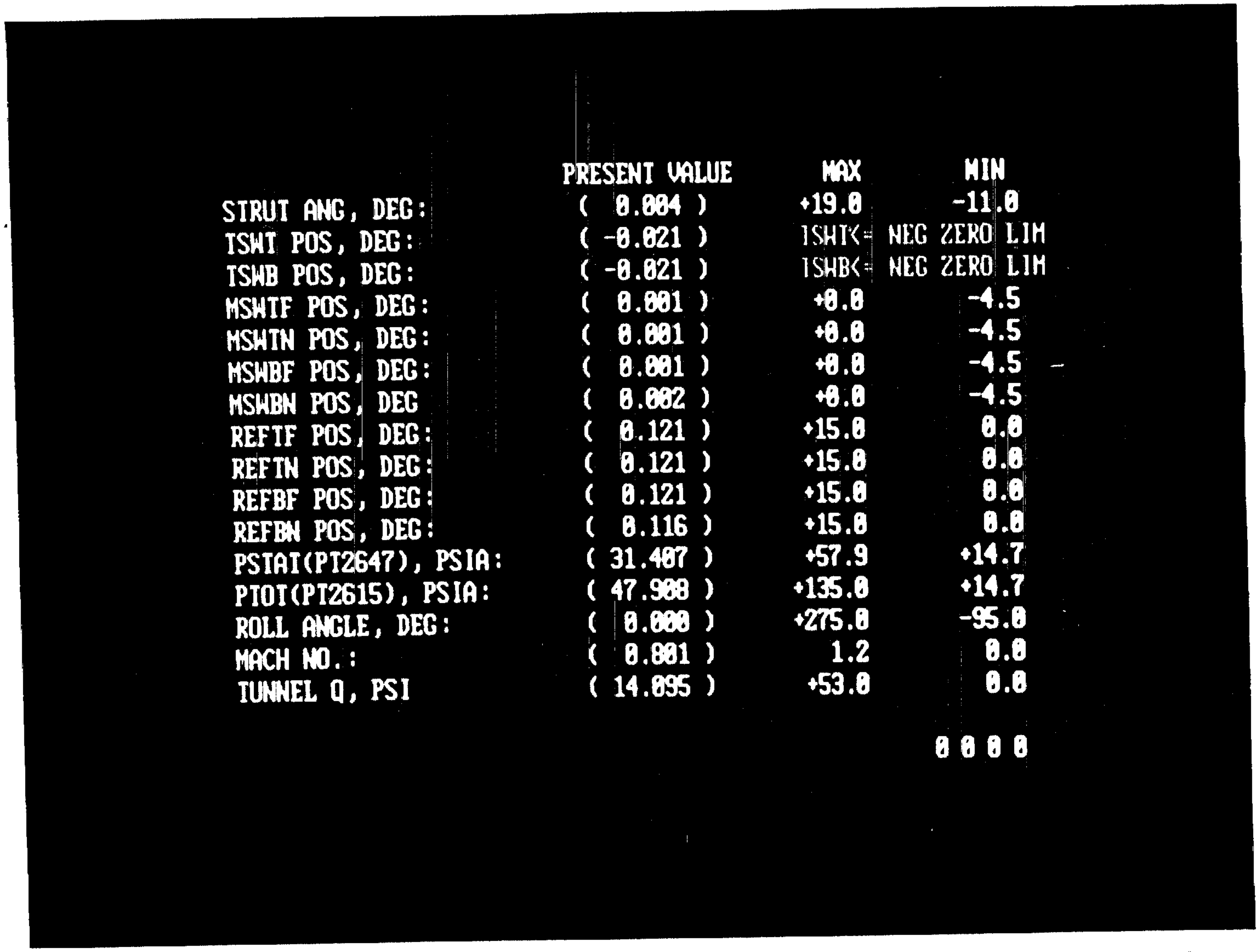
```
                           PRESENT VALUE      MAX        MIN
STRUT ANG, DEG:            ( -0.001 )        +19.0      -11.0
TSMT POS, DEG:             ( -0.002 )         +1.0       -0.5
TSMB POS, DEG:             ( -0.000 )         +1.0       -0.5
MSMTF POS, DEG:            (  0.001 )         +0.0       -4.5
MSMTN POS, DEG:            (  0.002 )         +0.0       -4.5
MSMBF POS, DEG:            (  0.001 )         +0.0       -4.5
MSMBN POS, DEG             (  0.002 )         +0.0       -4.5
REFTF POS, DEG:            (  0.118 )        +15.0        0.0
REFTN POS, DEG:            (  0.121 )        +15.0        0.0
REFBF POS, DEG:            (  0.122 )        +15.0        0.0
REFBN POS, DEG:            (  0.120 )        +15.0        0.0
PSTAT(PT2647), PSIA:       ( 31.413 )        +57.9      +14.7
PTOT(PT2615), PSIA:        ( 47.899 )       +135.0      +14.7
ROLL ANGLE, DEG:           (  0.000 )       +275.0      -95.0
MACH NO.:                  (  0.800 )          1.2        0.0
TUNNEL Q, PSI              ( 14.084 )        +53.0        0.0

                                                      0 0 0 0
```

Figure 4. - Basic display in unalarmed state.



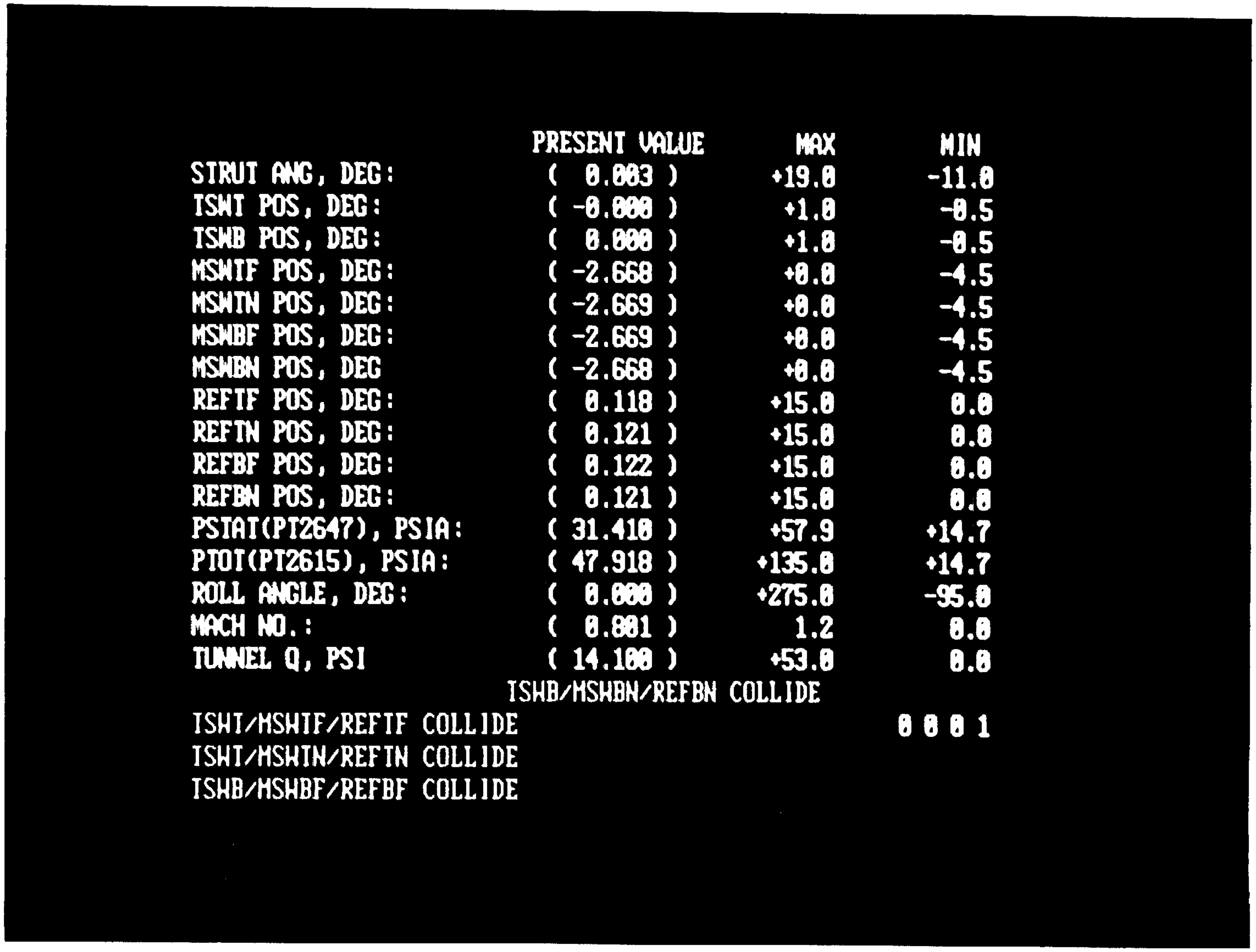
```
                          PRESENT VALUE     MAX          MIN
STRUT ANG, DEG:           ( -0.001 )       +19.0       -11.0
TSWT POS, DEG:            (  0.021 )       TSWT>= PLUS ZERO LIM
TSWB POS, DEG:            (  0.021 )       TSWB>= PLUS ZERO LIM
MSWTF POS, DEG:           (  0.001 )        +0.0        -4.5
MSWTN POS, DEG:           (  0.002 )        +0.0        -4.5
MSWBF POS, DEG:           (  0.001 )        +0.0        -4.5
MSWBN POS, DEG            (  0.001 )        +0.0        -4.5
REFTF POS, DEG:           (  0.121 )       +15.0         0.0
REFTN POS, DEG:           (  0.121 )       +15.0         0.0
REFBF POS, DEG:           (  0.120 )       +15.0         0.0
REFBN POS, DEG:           (  0.119 )       +15.0         0.0
PSTAT(PT2647), PSIA:      ( 31.410 )       +57.9       +14.7
PTOT(PT2615), PSIA:       ( 47.908 )      +135.0       +14.7
ROLL ANGLE, DEG:          (  0.000 )      +275.0       -95.0
MACH NO.:                 (  0.001 )         1.2         0.0
TUNNEL Q, PSI             ( 14.093 )       +53.0         0.0

                                                     0 0 0 0
```

Figure 5. - Test section top and bottom walls relative to positive limit position (TEST 01).



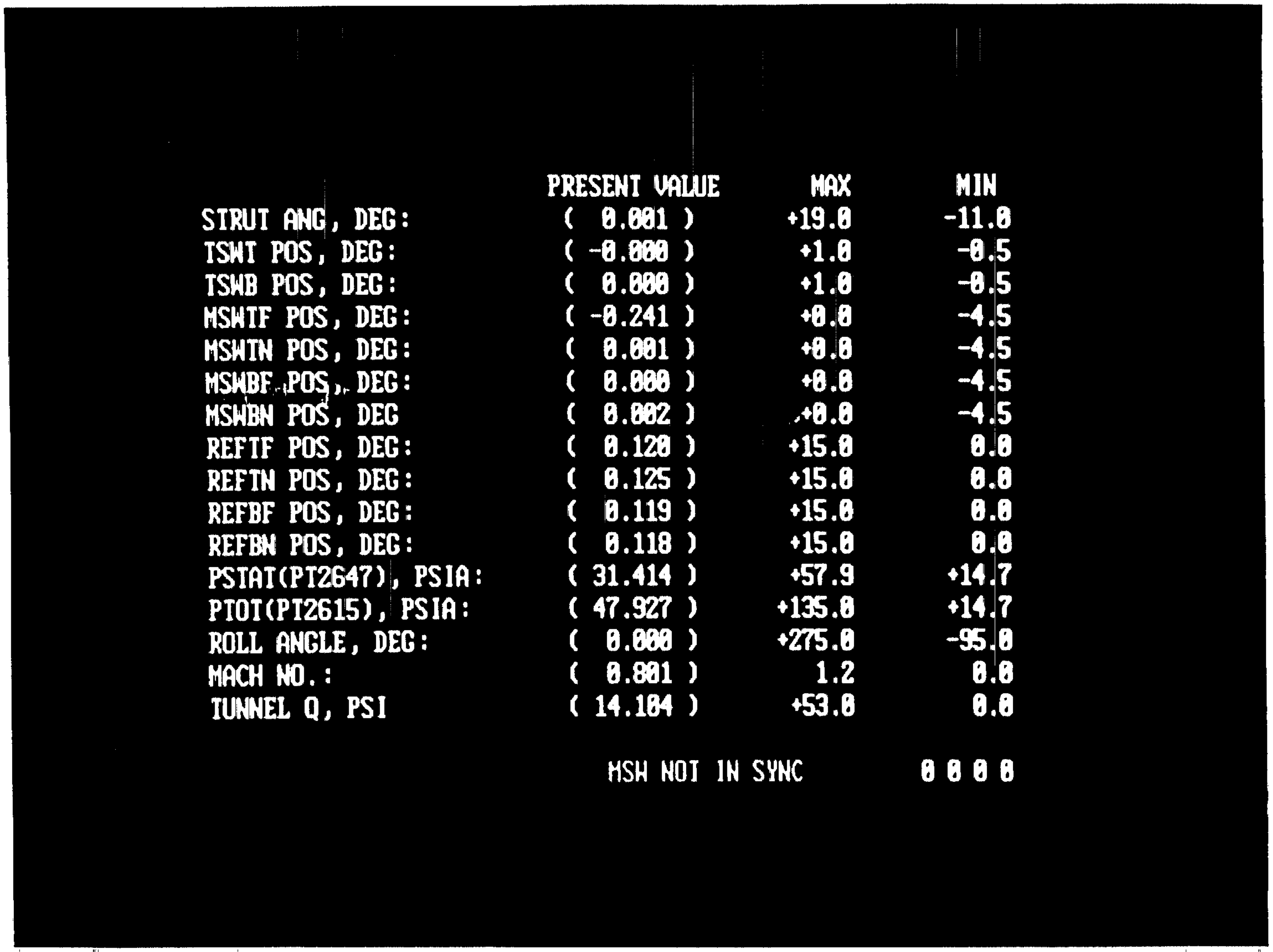
```
                           PRESENT VALUE      MAX        MIN
STRUT ANG, DEG:             (  0.004 )       +19.0       -11.0
TSWT POS, DEG:              ( -0.021 )       TSWT<= NEG ZERO LIM
TSWB POS, DEG:              ( -0.021 )       TSWB<= NEG ZERO LIM
MSWTF POS, DEG:             (  0.001 )        +0.0        -4.5
MSWTN POS, DEG:             (  0.001 )        +0.0        -4.5
MSWBF POS, DEG:             (  0.001 )        +0.0        -4.5
MSWBN POS, DEG              (  0.002 )        +0.0        -4.5
REFTF POS, DEG:             (  0.121 )       +15.0         0.0
REFTN POS, DEG:             (  0.121 )       +15.0         0.0
REFBF POS, DEG:             (  0.121 )       +15.0         0.0
REFBN POS, DEG:             (  0.116 )       +15.0         0.0
PSTAT(PT2647), PSIA:        ( 31.407 )       +57.9       +14.7
PTOT(PT2615), PSIA:         ( 47.908 )      +135.0       +14.7
ROLL ANGLE, DEG:            (  0.000 )      +275.0       -95.0
MACH NO.:                   (  0.001 )         1.2         0.0
TUNNEL Q, PSI               ( 14.095 )       +53.0         0.0

                                                     0 0 0 0
```

Figure 6. - Test section top and bottom walls relative to negative limit position (TEST 01).



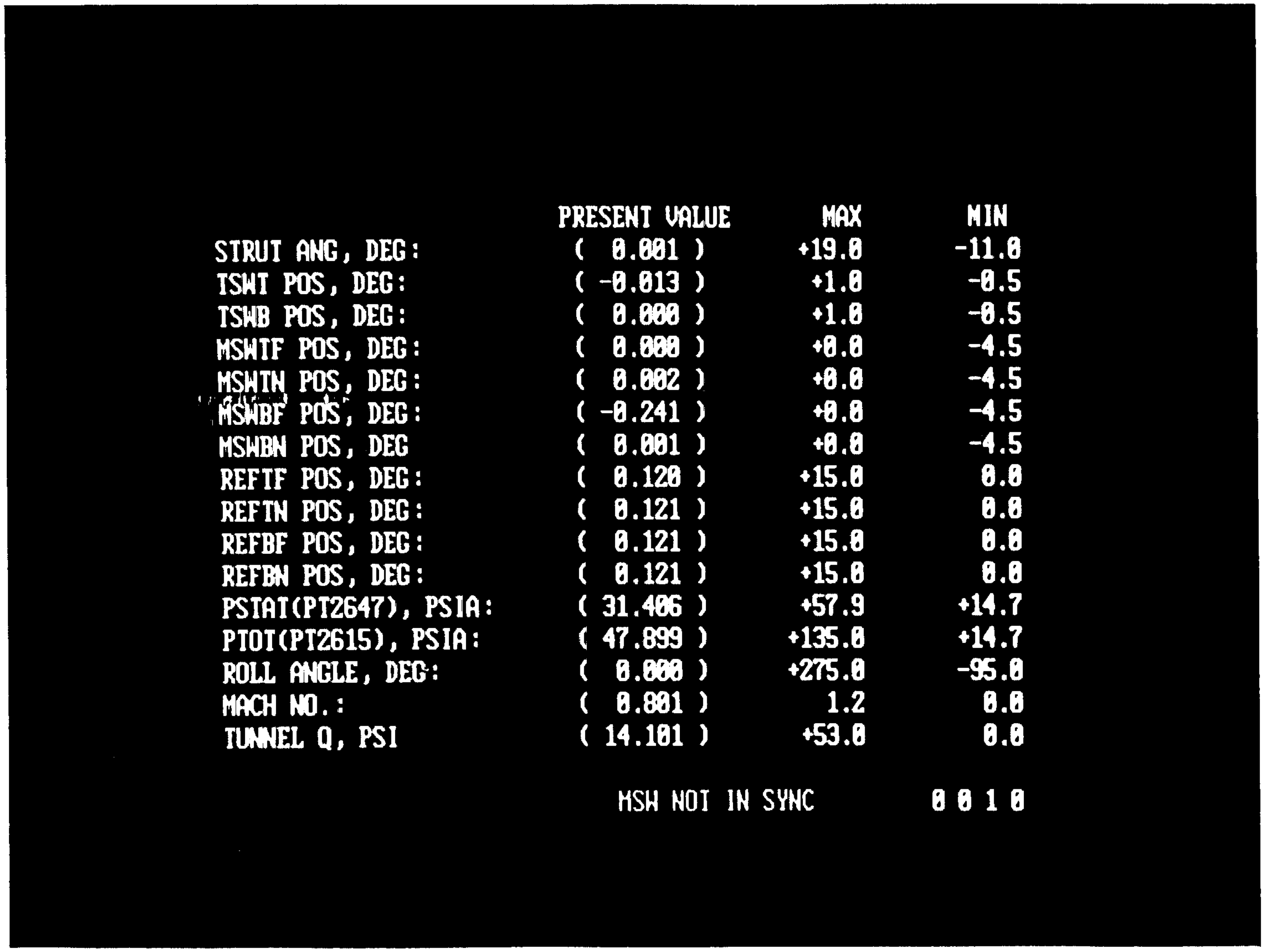
```
                                PRESENT VALUE      MAX       MIN
STRUT ANG, DEG:                 (  0.003 )        +19.0     -11.0
ISWT POS, DEG:                  ( -0.000 )         +1.0      -0.5
ISWB POS, DEG:                  (  0.000 )         +1.0      -0.5
MSWTF POS, DEG:                 ( -2.668 )         +0.0      -4.5
MSWTN POS, DEG:                 ( -2.669 )         +0.0      -4.5
MSWBF POS, DEG:                 ( -2.669 )         +0.0      -4.5
MSWBN POS, DEG                  ( -2.668 )         +0.0      -4.5
REFTF POS, DEG:                 (  0.118 )        +15.0       0.0
REFTN POS, DEG:                 (  0.121 )        +15.0       0.0
REFBF POS, DEG:                 (  0.122 )        +15.0       0.0
REFBN POS, DEG:                 (  0.121 )        +15.0       0.0
PSTAT(PT2647), PSIA:            ( 31.410 )        +57.9     +14.7
PTOT(PT2615), PSIA:             ( 47.918 )       +135.0     +14.7
ROLL ANGLE, DEG:                (  0.000 )       +275.0     -95.0
MACH NO.:                       (  0.881 )          1.2       0.0
TUNNEL Q, PSI                   ( 14.100 )        +53.0       0.0
                              ISWB/MSWBN/REFBN COLLIDE
ISWT/MSWTF/REFTF COLLIDE                                  0 0 0 1
ISWT/MSWTN/REFTN COLLIDE
ISWB/MSWBF/REFBF COLLIDE
```

Figure 7. - Top and bottom test section walls relative to model support walls and reentry flaps (TEST 02).



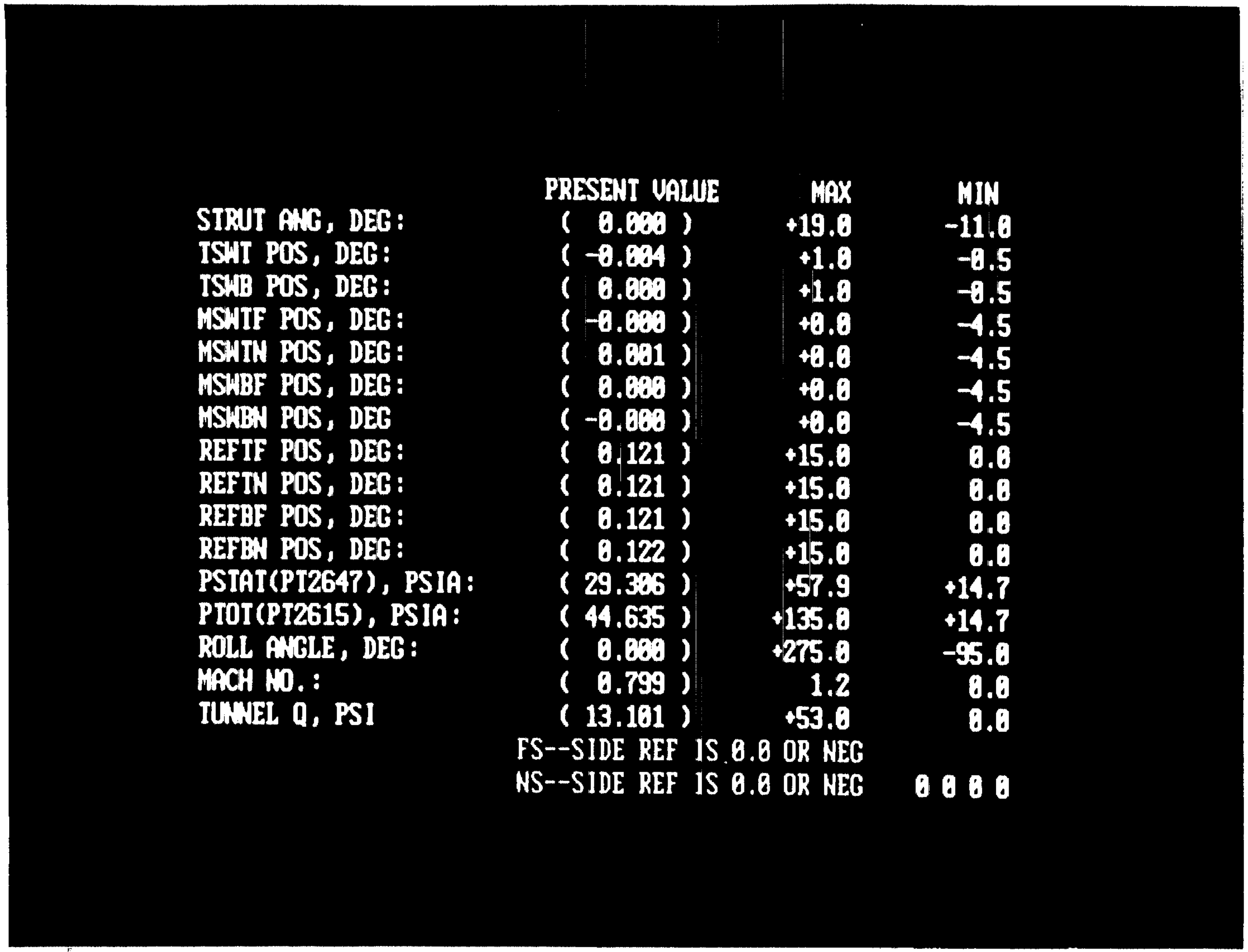
```
                         PRESENT VALUE      MAX       MIN
STRUT ANG, DEG:          (  0.001 )        +19.0     -11.0
TSWT POS, DEG:           ( -0.000 )         +1.0      -0.5
TSWB POS, DEG:           (  0.000 )         +1.0      -0.5
MSWTF POS, DEG:          ( -0.241 )         +0.0      -4.5
MSWTN POS, DEG:          (  0.001 )         +0.0      -4.5
MSWBF POS, DEG:          (  0.000 )         +0.0      -4.5
MSWBN POS, DEG           (  0.002 )         +0.0      -4.5
REFTF POS, DEG:          (  0.120 )        +15.0       0.0
REFTN POS, DEG:          (  0.125 )        +15.0       0.0
REFBF POS, DEG:          (  0.119 )        +15.0       0.0
REFBN POS, DEG:          (  0.118 )        +15.0       0.0
PSTAT(PT2647), PSIA:     ( 31.414 )        +57.9     +14.7
PTOT(PT2615), PSIA:      ( 47.927 )       +135.0     +14.7
ROLL ANGLE, DEG:         (  0.000 )       +275.0     -95.0
MACH NO.:                (  0.801 )          1.2       0.0
TUNNEL Q, PSI            ( 14.104 )        +53.0       0.0

                             MSW NOT IN SYNC       0 0 0 0
```

Figure 8. - Top model support walls synchronization (TEST 03).



```
                         PRESENT VALUE      MAX        MIN
STRUT ANG, DEG:           (  0.001 )       +19.0      -11.0
TSWT POS, DEG:            ( -0.013 )        +1.0       -0.5
TSWB POS, DEG:            (  0.000 )        +1.0       -0.5
MSWTF POS, DEG:           (  0.000 )        +0.0       -4.5
MSWTN POS, DEG:           (  0.002 )        +0.0       -4.5
MSWBF POS, DEG:           ( -0.241 )        +0.0       -4.5
MSWBN POS, DEG            (  0.001 )        +0.0       -4.5
REFTF POS, DEG:           (  0.120 )       +15.0        0.0
REFTN POS, DEG:           (  0.121 )       +15.0        0.0
REFBF POS, DEG:           (  0.121 )       +15.0        0.0
REFBN POS, DEG:           (  0.121 )       +15.0        0.0
PSTAT(PT2647), PSIA:      ( 31.406 )       +57.9      +14.7
PTOT(PT2615), PSIA:       ( 47.899 )      +135.0      +14.7
ROLL ANGLE, DEG:          (  0.000 )      +275.0      -95.0
MACH NO.:                 (  0.801 )         1.2        0.0
TUNNEL Q, PSI             ( 14.101 )       +53.0        0.0

                              MSW NOT IN SYNC       0 0 1 0
```

Figure 9. - Bottom model support walls synchronization (TEST 03).



```
                        PRESENT VALUE       MAX        MIN
STRUT ANG, DEG:          (  0.000 )        +19.0      -11.0
TSWT POS, DEG:           ( -0.004 )         +1.0       -0.5
TSWB POS, DEG:           (  0.000 )         +1.0       -0.5
MSWTF POS, DEG:          ( -0.000 )         +0.0       -4.5
MSWTN POS, DEG:          (  0.001 )         +0.0       -4.5
MSWBF POS, DEG:          (  0.000 )         +0.0       -4.5
MSWBN POS, DEG           ( -0.000 )         +0.0       -4.5
REFTF POS, DEG:          (  0.121 )        +15.0        0.0
REFTN POS, DEG:          (  0.121 )        +15.0        0.0
REFBF POS, DEG:          (  0.121 )        +15.0        0.0
REFBN POS, DEG:          (  0.122 )        +15.0        0.0
PSTAT(PT2647), PSIA:     ( 29.306 )        +57.9      +14.7
PTOT(PT2615), PSIA:      ( 44.635 )       +135.0      +14.7
ROLL ANGLE, DEG:         (  0.000 )       +275.0      -95.0
MACH NO.:                (  0.799 )          1.2        0.0
TUNNEL Q, PSI            ( 13.101 )        +53.0        0.0
                       FS--SIDE REF IS 0.0 OR NEG
                       NS--SIDE REF IS 0.0 OR NEG    0 0 0 0
```

Figure 10. - Side reentry flaps relative to zero limit (TEST 05).

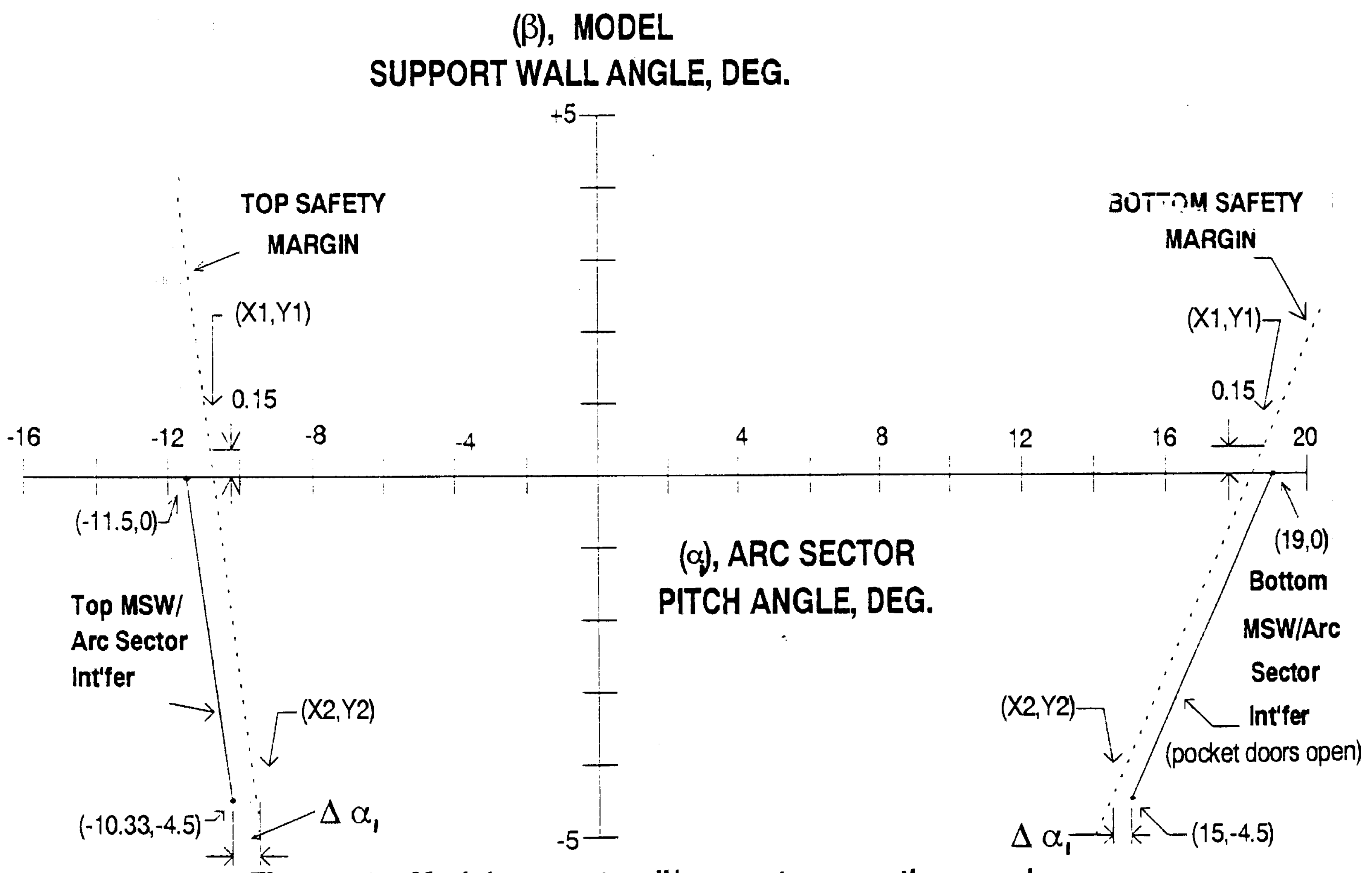


Figure 11. - Model support wall/arc sector operating envelope.

WALL- STRUT INTERFERENCE
INEQUALITIES

$$4.5\,\alpha_1 + 1.17\,\beta >= -51.5\overset{1}{\cancel{8}}$$

$$4.5\,\alpha_1 - 4\beta <= 84.9$$

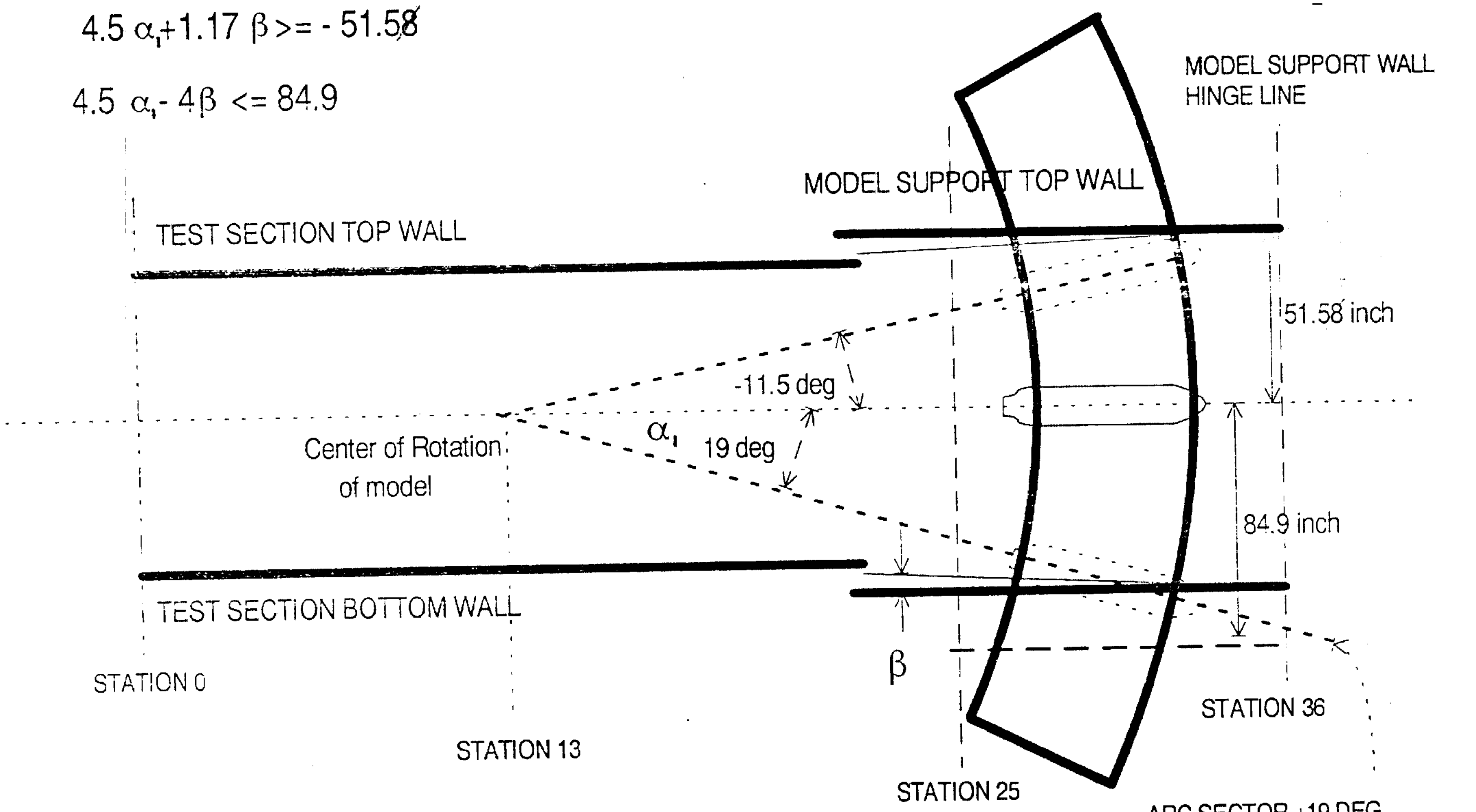
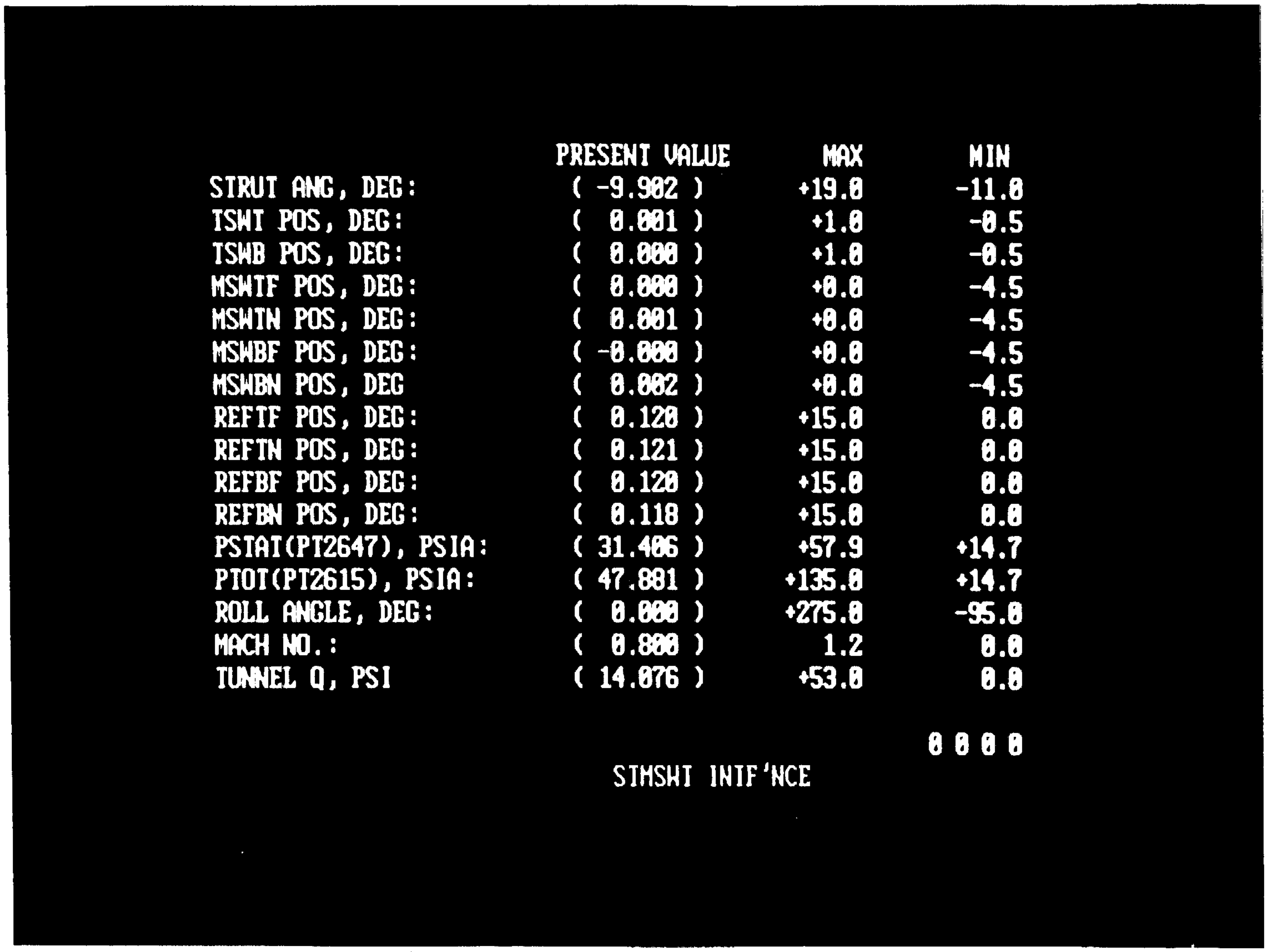
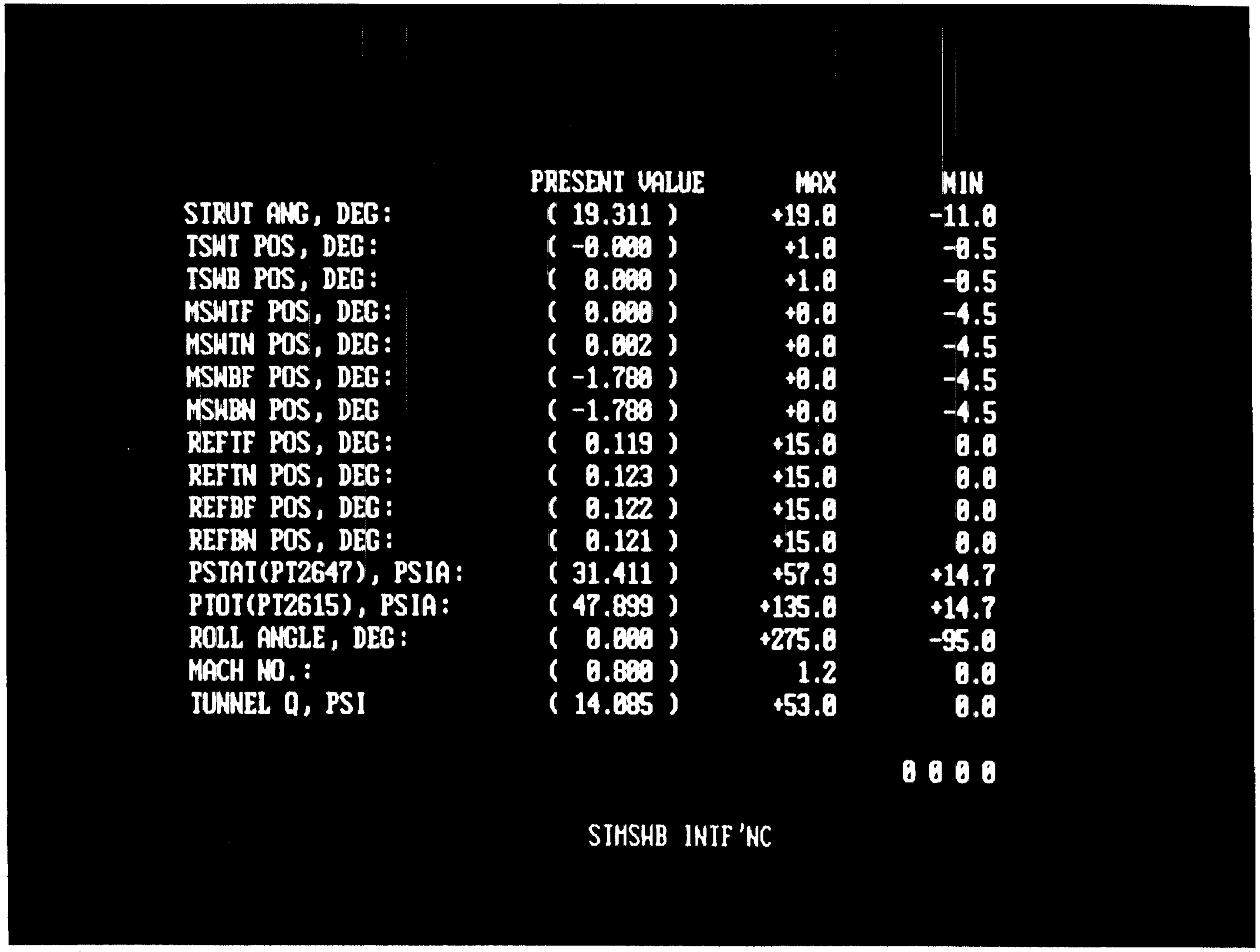


Figure 12.- Model support wall/arc sector interference limits.



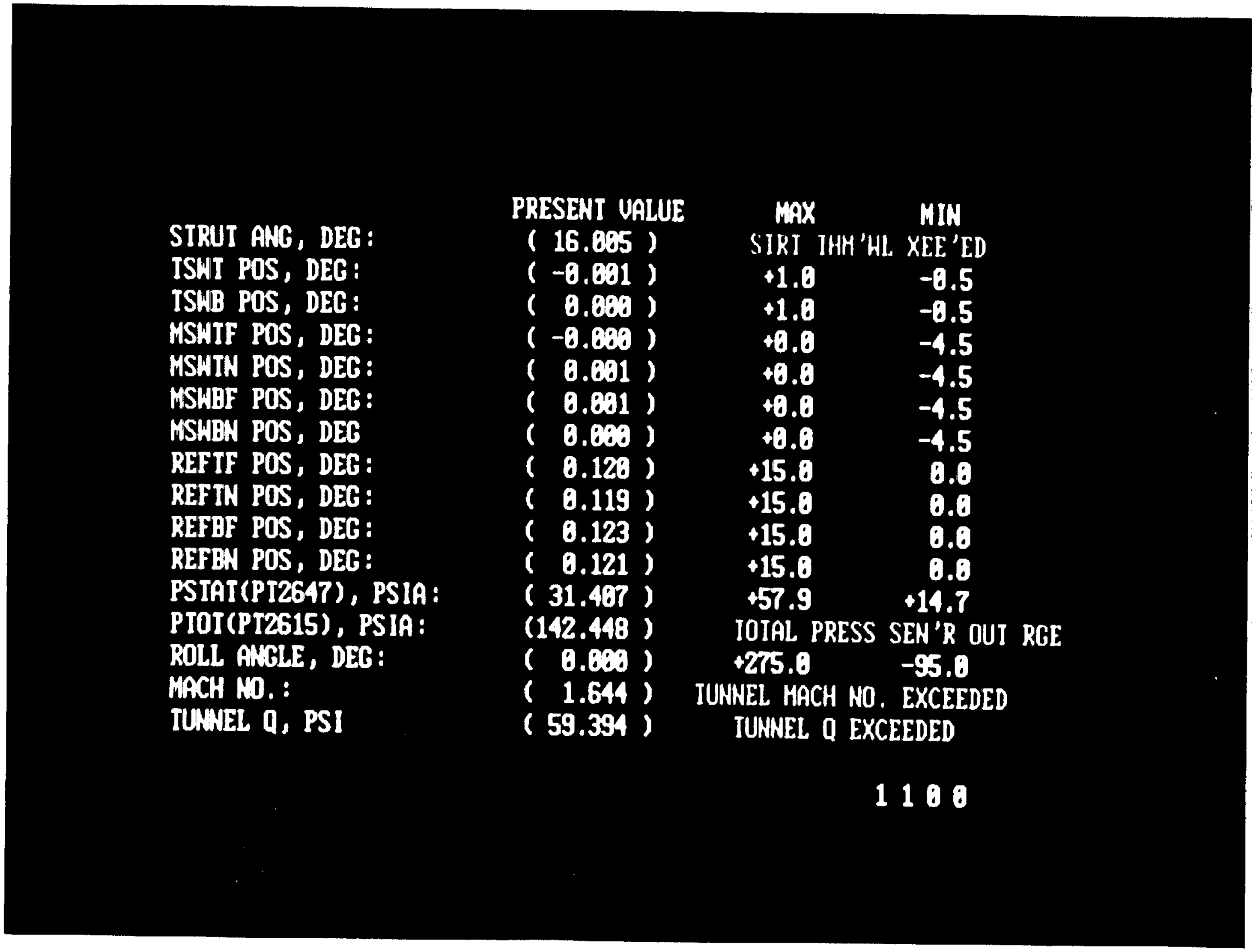
```
                           PRESENT VALUE      MAX        MIN
STRUT ANG, DEG:             ( -9.902 )       +19.0      -11.0
TSWT POS, DEG:              (  0.001 )        +1.0       -0.5
TSWB POS, DEG:              (  0.000 )        +1.0       -0.5
MSWTF POS, DEG:             (  0.000 )        +0.0       -4.5
MSWTN POS, DEG:             (  0.001 )        +0.0       -4.5
MSWBF POS, DEG:             ( -0.000 )        +0.0       -4.5
MSWBN POS, DEG              (  0.002 )        +0.0       -4.5
REFTF POS, DEG:             (  0.120 )       +15.0        0.0
REFTN POS, DEG:             (  0.121 )       +15.0        0.0
REFBF POS, DEG:             (  0.120 )       +15.0        0.0
REFBN POS, DEG:             (  0.118 )       +15.0        0.0
PSTAT(PT2647), PSIA:        ( 31.406 )       +57.9      +14.7
PTOT(PT2615), PSIA:         ( 47.881 )      +135.0      +14.7
ROLL ANGLE, DEG:            (  0.000 )      +275.0      -95.0
MACH NO.:                   (  0.800 )         1.2        0.0
TUNNEL Q, PSI               ( 14.076 )       +53.0        0.0

                                                      0 0 0 0
                             STMSWT INTF'NCE
```

Figure 13. - Top model support wall/arc sector interference (TEST 06).



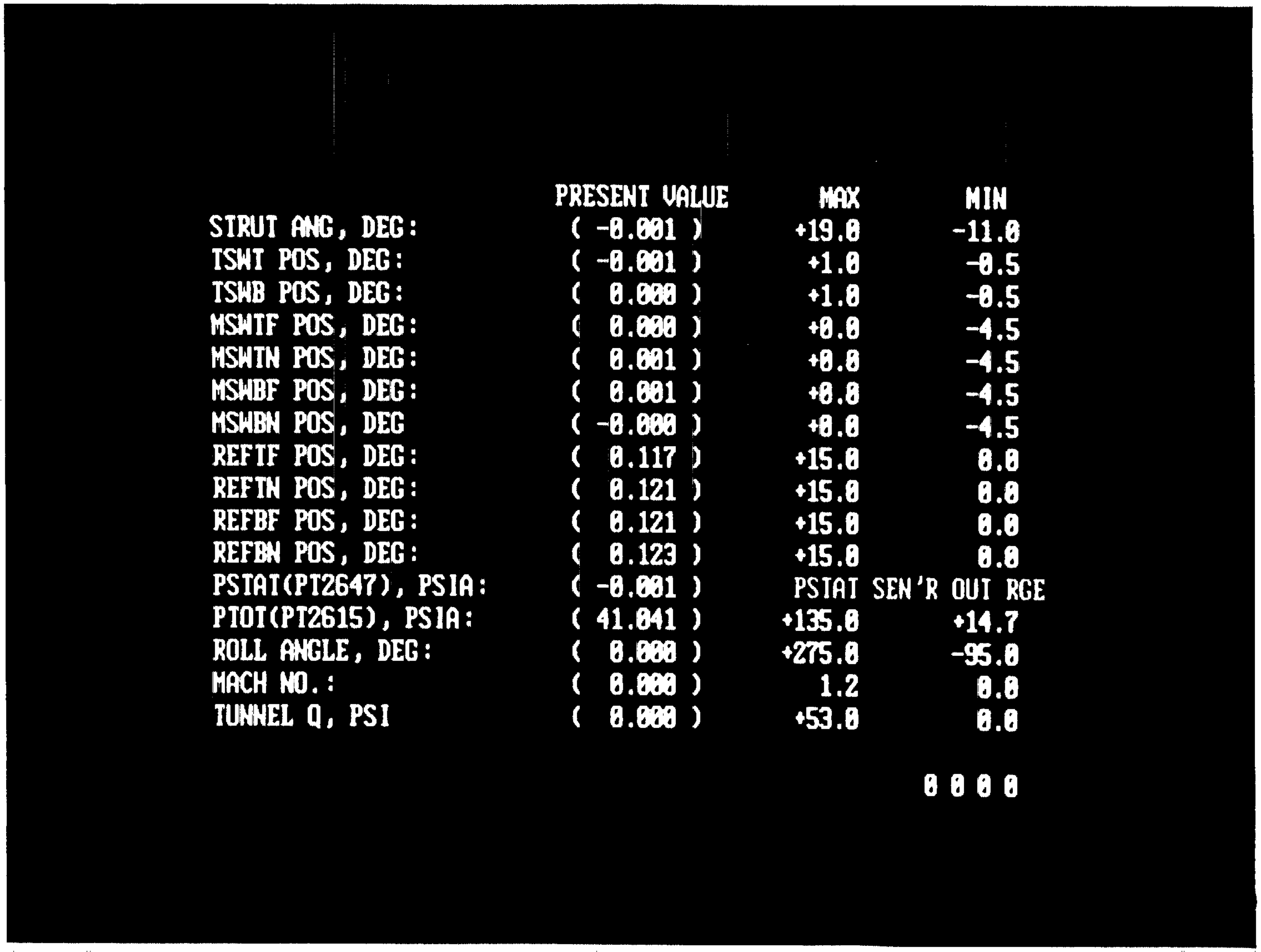
```
                        PRESENT VALUE      MAX        MIN
STRUT ANG, DEG:          ( 19.311 )      +19.0      -11.0
TSWT POS, DEG:           ( -0.000 )       +1.0       -0.5
TSWB POS, DEG:           (  0.000 )       +1.0       -0.5
MSWTF POS, DEG:          (  0.000 )       +0.0       -4.5
MSWTN POS, DEG:          (  0.002 )       +0.0       -4.5
MSWBF POS, DEG:          ( -1.780 )       +0.0       -4.5
MSWBN POS, DEG           ( -1.780 )       +0.0       -4.5
REFTF POS, DEG:          (  0.119 )      +15.0        0.0
REFTN POS, DEG:          (  0.123 )      +15.0        0.0
REFBF POS, DEG:          (  0.122 )      +15.0        0.0
REFBN POS, DEG:          (  0.121 )      +15.0        0.0
PSTAT(PT2647), PSIA:     ( 31.411 )      +57.9      +14.7
PTOT(PT2615), PSIA:      ( 47.899 )     +135.0      +14.7
ROLL ANGLE, DEG:         (  0.000 )     +275.0      -95.0
MACH NO.:                (  0.000 )        1.2        0.0
TUNNEL Q, PSI            ( 14.085 )      +53.0        0.0

                                                  0 0 0 0

                             SIMSHB INTF'NC
```

Figure 14. - Bottom model support wall/arc sector interference (TEST 06).



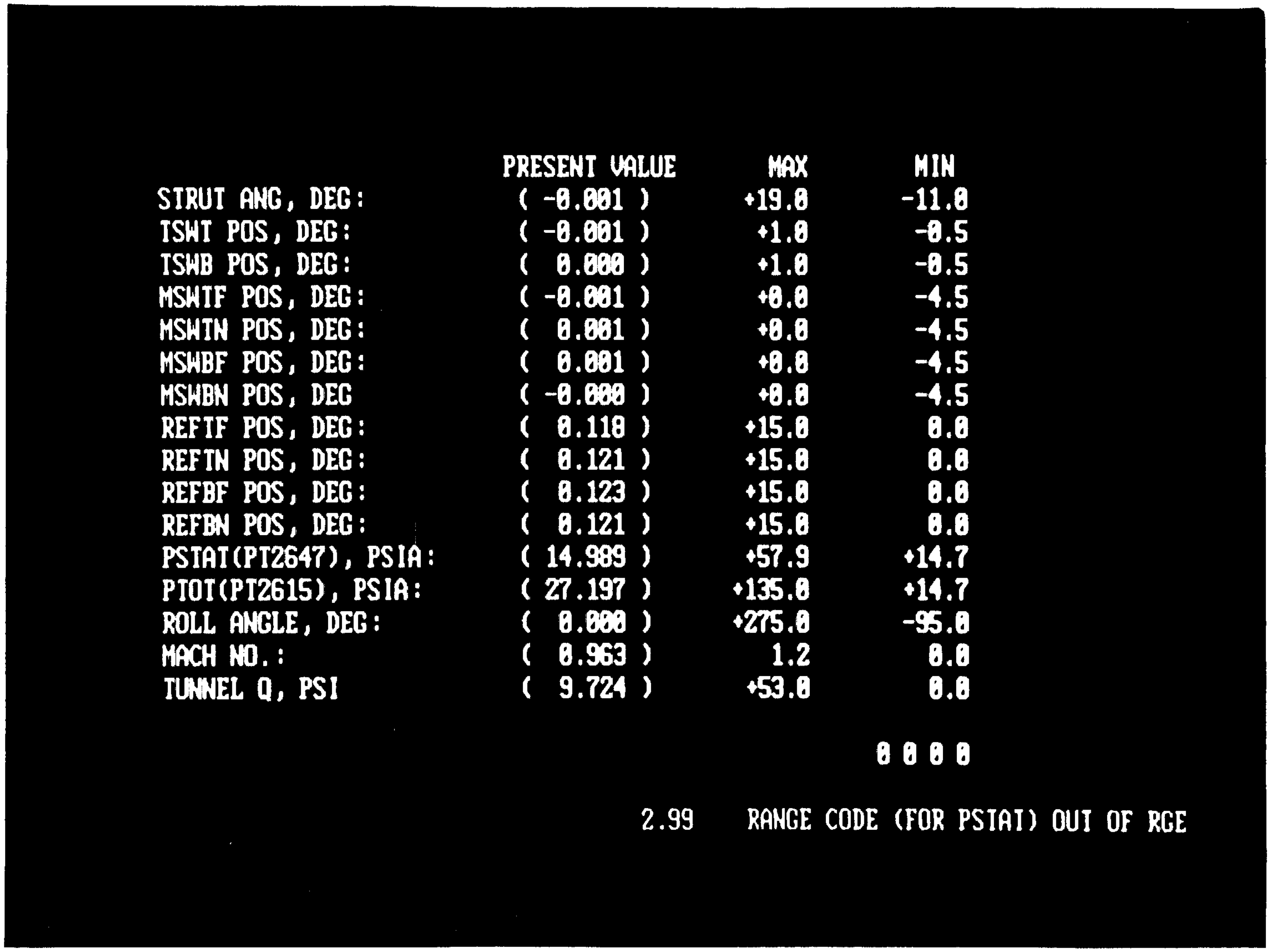
```
                            PRESENT VALUE      MAX        MIN
STRUT ANG, DEG:              ( 16.005 )      SIRT IHH'HL XEE'ED
ISWT POS, DEG:               ( -0.001 )       +1.0       -0.5
TSWB POS, DEG:               (  0.000 )       +1.0       -0.5
MSWTF POS, DEG:              ( -0.000 )       +0.0       -4.5
MSWTN POS, DEG:              (  0.001 )       +0.0       -4.5
MSWBF POS, DEG:              (  0.001 )       +0.0       -4.5
MSWBN POS, DEG               (  0.000 )       +0.0       -4.5
REFTF POS, DEG:              (  0.120 )      +15.0        0.0
REFTN POS, DEG:              (  0.119 )      +15.0        0.0
REFBF POS, DEG:              (  0.123 )      +15.0        0.0
REFBN POS, DEG:              (  0.121 )      +15.0        0.0
PSTAT(PT2647), PSIA:         ( 31.407 )      +57.9      +14.7
PTOT(PT2615), PSIA:          (142.448 )     IOIAL PRESS SEN'R OUI RGE
ROLL ANGLE, DEG:             (  0.000 )     +275.0      -95.0
MACH NO.:                    (  1.644 )   IUNNEL MACH NO. EXCEEDED
TUNNEL Q, PSI                ( 59.394 )     IUNNEL Q EXCEEDED

                                                     1 1 0 0
```

Figure 15. - Additional alerts (TEST 07).



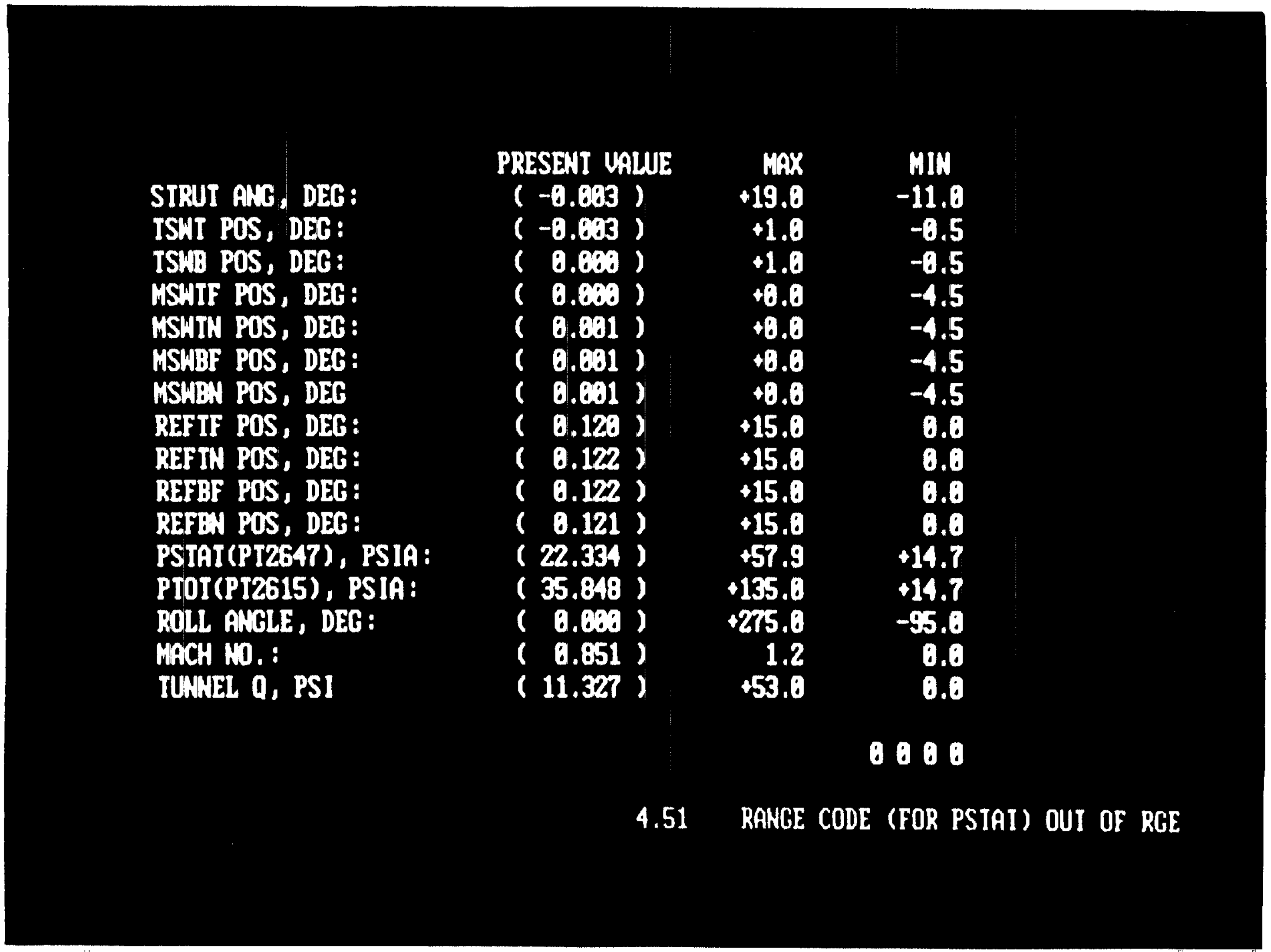
```
                           PRESENT VALUE     MAX        MIN
STRUT ANG, DEG:             ( -0.001 )      +19.0      -11.0
TSWT POS, DEG:              ( -0.001 )       +1.0       -0.5
TSWB POS, DEG:              (  0.000 )       +1.0       -0.5
MSWTF POS, DEG:             (  0.000 )       +0.0       -4.5
MSWTN POS, DEG:             (  0.001 )       +0.0       -4.5
MSWBF POS, DEG:             (  0.001 )       +0.0       -4.5
MSWBN POS, DEG              ( -0.000 )       +0.0       -4.5
REFTF POS, DEG:             (  0.117 )      +15.0        0.0
REFTN POS, DEG:             (  0.121 )      +15.0        0.0
REFBF POS, DEG:             (  0.121 )      +15.0        0.0
REFBN POS, DEG:             (  0.123 )      +15.0        0.0
PSTAT(PT2647), PSIA:        ( -0.001 )      PSTAT SEN'R OUT RGE
PTOT(PT2615), PSIA:         ( 41.041 )     +135.0      +14.7
ROLL ANGLE, DEG:            (  0.000 )     +275.0      -95.0
MACH NO.:                   (  0.000 )        1.2        0.0
TUNNEL Q, PSI               (  0.000 )      +53.0        0.0

                                                   0 0 0 0
```

Figure 16. - Additional alerts continued (TEST 07).



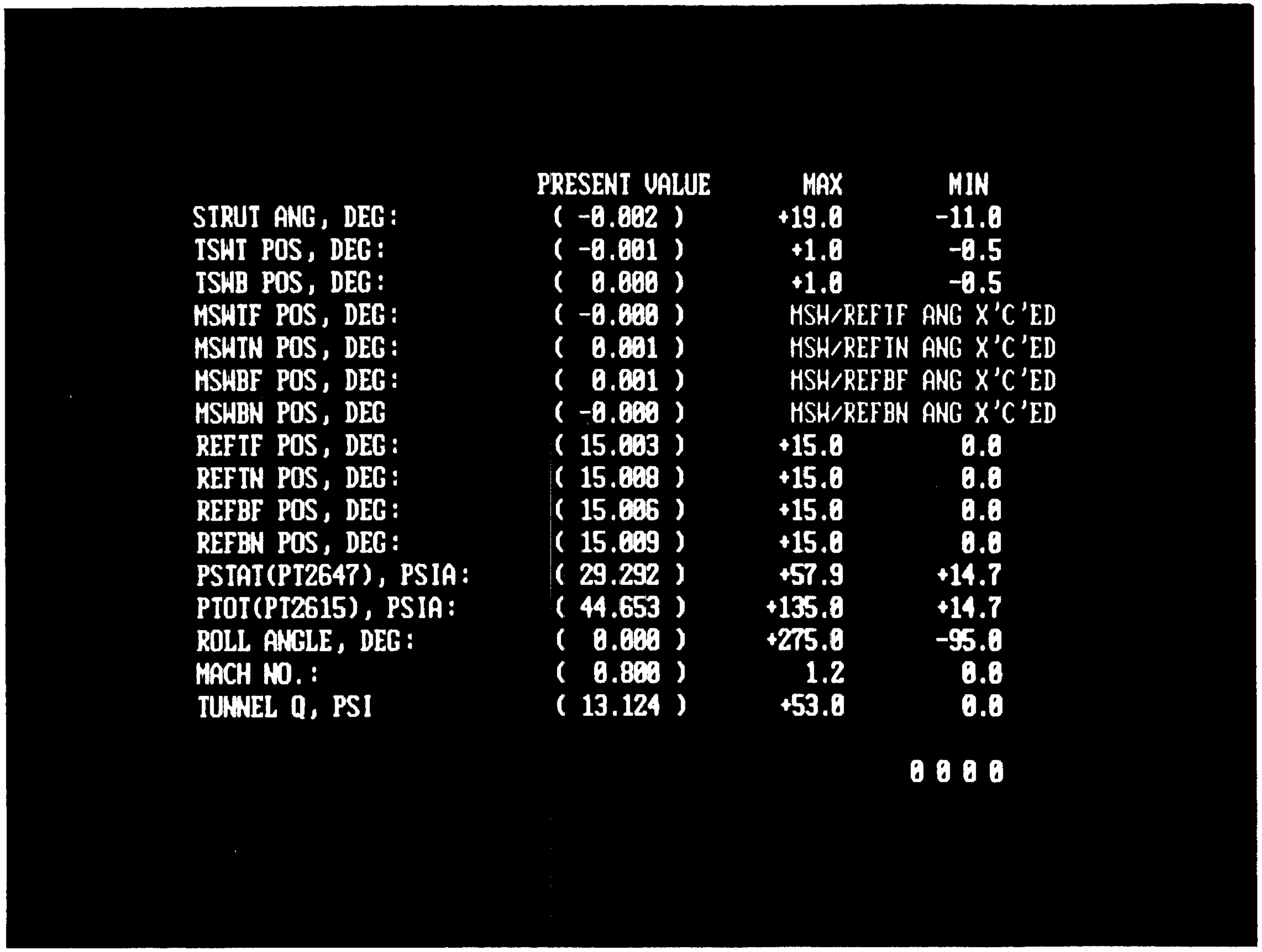
```
                          PRESENT VALUE     MAX       MIN
STRUT ANG, DEG:           ( -0.001 )       +19.0     -11.0
TSWT POS, DEG:            ( -0.001 )        +1.0      -0.5
TSWB POS, DEG:            (  0.000 )        +1.0      -0.5
MSWTF POS, DEG:           ( -0.001 )        +0.0      -4.5
MSWTN POS, DEG:           (  0.001 )        +0.0      -4.5
MSWBF POS, DEG:           (  0.001 )        +0.0      -4.5
MSWBN POS, DEG            ( -0.000 )        +0.0      -4.5
REFTF POS, DEG:           (  0.118 )       +15.0       0.0
REFTN POS, DEG:           (  0.121 )       +15.0       0.0
REFBF POS, DEG:           (  0.123 )       +15.0       0.0
REFBN POS, DEG:           (  0.121 )       +15.0       0.0
PSTAT(PT2647), PSIA:      ( 14.989 )       +57.9     +14.7
PTOT(PT2615), PSIA:       ( 27.197 )      +135.0     +14.7
ROLL ANGLE, DEG:          (  0.000 )      +275.0     -95.0
MACH NO.:                 (  0.963 )         1.2       0.0
TUNNEL Q, PSI             (  9.724 )       +53.0       0.0

                                                  0 0 0 0

                                   2.99   RANGE CODE (FOR PSTAT) OUT OF RGE
```

Figure 17. - Additional alerts continued (TEST 07).



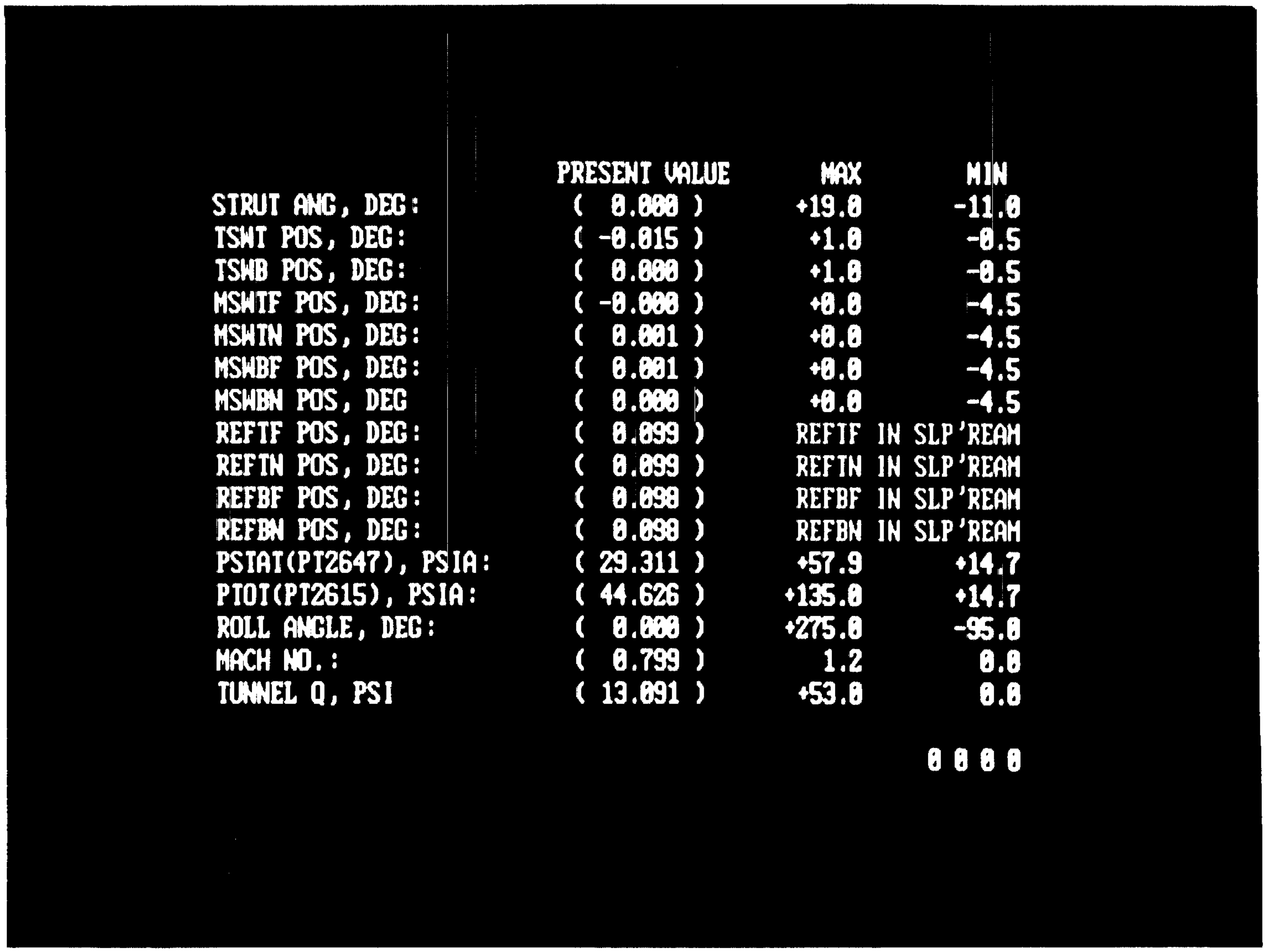
```
                        PRESENT VALUE      MAX       MIN
STRUT ANG, DEG:          ( -0.003 )      +19.0     -11.0
TSWT POS, DEG:           ( -0.003 )       +1.0      -0.5
TSWB POS, DEG:           (  0.000 )       +1.0      -0.5
MSWTF POS, DEG:          (  0.000 )       +0.0      -4.5
MSWTN POS, DEG:          (  0.001 )       +0.0      -4.5
MSWBF POS, DEG:          (  0.001 )       +0.0      -4.5
MSWBN POS, DEG           (  0.001 )       +0.0      -4.5
REFTF POS, DEG:          (  0.120 )      +15.0       0.0
REFTN POS, DEG:          (  0.122 )      +15.0       0.0
REFBF POS, DEG:          (  0.122 )      +15.0       0.0
REFBN POS, DEG:          (  0.121 )      +15.0       0.0
PSTAT(PT2647), PSIA:     ( 22.334 )      +57.9     +14.7
PTOT(PT2615), PSIA:      ( 35.848 )     +135.0     +14.7
ROLL ANGLE, DEG:         (  0.000 )     +275.0     -95.0
MACH NO.:                (  0.851 )        1.2       0.0
TUNNEL Q, PSI            ( 11.327 )      +53.0       0.0

                                               0 0 0 0

                            4.51    RANGE CODE (FOR PSTAT) OUT OF RGE
```

Figure 18. - Additional alerts concluded (TEST 07).



```
                          PRESENT VALUE      MAX        MIN
STRUT ANG, DEG:           ( -0.002 )       +19.0      -11.0
TSWT POS, DEG:            ( -0.001 )        +1.0       -0.5
TSWB POS, DEG:            (  0.000 )        +1.0       -0.5
MSWTF POS, DEG:           ( -0.000 )       MSW/REFTF ANG X'C'ED
MSWTN POS, DEG:           (  0.001 )       MSW/REFTN ANG X'C'ED
MSWBF POS, DEG:           (  0.001 )       MSW/REFBF ANG X'C'ED
MSWBN POS, DEG            ( -0.000 )       MSW/REFBN ANG X'C'ED
REFTF POS, DEG:           ( 15.003 )       +15.0        0.0
REFTN POS, DEG:           ( 15.008 )       +15.0        0.0
REFBF POS, DEG:           ( 15.006 )       +15.0        0.0
REFBN POS, DEG:           ( 15.009 )       +15.0        0.0
PSTAT(PT2647), PSIA:      ( 29.292 )       +57.9      +14.7
PTOT(PT2615), PSIA:       ( 44.653 )      +135.0      +14.7
ROLL ANGLE, DEG:          (  0.000 )      +275.0      -95.0
MACH NO.:                 (  0.800 )         1.2        0.0
TUNNEL Q, PSI             ( 13.124 )       +53.0        0.0

                                                    0 0 0 0
```

Figure 19. - Top and bottom model support walls to reentry flap angular position (TEST 10).



```
                        PRESENT VALUE      MAX        MIN
STRUT ANG, DEG:         (  0.000 )       +19.0      -11.0
TSWT POS, DEG:          ( -0.015 )        +1.0       -0.5
TSWB POS, DEG:          (  0.000 )        +1.0       -0.5
MSWTF POS, DEG:         ( -0.000 )        +0.0       -4.5
MSWTN POS, DEG:         (  0.001 )        +0.0       -4.5
MSWBF POS, DEG:         (  0.001 )        +0.0       -4.5
MSWBN POS, DEG          (  0.000 )        +0.0       -4.5
REFTF POS, DEG:         (  0.099 )       REFTF IN SLP'REAM
REFTN POS, DEG:         (  0.099 )       REFTN IN SLP'REAM
REFBF POS, DEG:         (  0.098 )       REFBF IN SLP'REAM
REFBN POS, DEG:         (  0.098 )       REFBN IN SLP'REAM
PSTAT(PT2647), PSIA:    ( 29.311 )       +57.9      +14.7
PTOT(PT2615), PSIA:     ( 44.626 )      +135.0      +14.7
ROLL ANGLE, DEG:        (  0.000 )      +275.0      -95.0
MACH NO.:               (  0.799 )         1.2        0.0
TUNNEL Q, PSI           ( 13.091 )       +53.0        0.0

                                                  0 0 0 0
```

Figure 20. - Top and bottom/near-side and far-side reentry flap angular position (TEST 10).

APPENDIX A

Simulation Source Code

```
'BEGIN MICRO 002 SIMULATION
'JAN 25, 1992
'THIS READS IN THE SYSTEM CONSTANTS AND VALUES FROM THE ANALOG-TO-DIGITAL
'CONVERTER AND RETURNS WITH A REAL SET OF DATA.
CLS
CLEAR
SCREEN 9
COLOR 14
'DEFINE THE A/D CONSTANTS
DIM VOLTS, A(8), B(8), C(8), DATA.VALUE(9)
DIM COSX(31), COSY(31), COSSLP(31), ISNX(13), ISNY(13), ISNSLP(13)

COMMAND.WAIT = &H4
WRITE.WAIT = &H2
READ.WAIT = &H5
CSTOP = &HF
CCLEAR = &H1
CADIN = &HC
'NUMBER OF CONVERSIONS
FACTOR# = 32768
RANGE = 10
GAIN = 1
GAIN.CODE = 0
SCALE = RANGE / FACTOR# / GAIN

'DEFINE "INITIAL" PROGRAM CONSTANTS
KCFS = .04256
KCL = 9.276
KLENBD = 132!
KLENA = 300!
KSTA36 = 432!
KBC = 60!
K180 = 180!
KCD1 = 21024!
KCD2 = 15840!
KQ1 = .7
KQ2 = 1!
KQ3 = .5
K45 = 4.5
K117 = 1.17
K40 = 4!
RANKIN = 459.67
COSX(1) = 0!
COSX(2) = 3!
COSX(3) = 6!
COSX(4) = 9!
COSX(5) = 12!
COSX(6) = 15!
COSX(7) = 18!
COSX(8) = 21!
COSX(9) = 24!
COSX(10) = 27!
COSX(11) = 30!
COSX(12) = 33!
COSX(13) = 36!
COSX(14) = 39!
COSX(15) = 42!
COSX(16) = 45!
COSX(17) = 48!
COSX(18) = 51!
COSX(19) = 54!
COSX(20) = 57!
COSX(21) = 60!
COSX(22) = 63!
COSX(23) = 66!
COSX(24) = 69!
COSX(25) = 72!
COSX(26) = 75!
COSX(27) = 78!
COSX(28) = 81!
COSX(29) = 84!
COSX(30) = 87!
COSX(31) = 1E+30

COSY(1) = 1!
COSY(2) = .99863
COSY(3) = .99452
COSY(4) = .98769
COSY(5) = .97815
COSY(6) = .96593
COSY(7) = .95106
COSY(8) = .93358
COSY(9) = .91355
COSY(10) = .89101
COSY(11) = .86603
COSY(12) = .83867
COSY(13) = .80902
COSY(14) = .77715
COSY(15) = .74314
COSY(16) = .70711
COSY(17) = .66913
COSY(18) = .62932
COSY(19) = .58779
COSY(20) = .54464
COSY(21) = .5
COSY(22) = .45399
COSY(23) = .40674
COSY(24) = .35837
COSY(25) = .30902
COSY(26) = .25882
COSY(27) = .20791
COSY(28) = .15645
COSY(29) = .10453
COSY(30) = .05234
COSY(31) = 1E+30

COSSLP(1) = -4.5682E-04
COSSLP(2) = -.0013692
COSSLP(3) = -.0022779
COSSLP(4) = -.0031802
COSSLP(5) = -.0040739
COSSLP(6) = -.0049564
COSSLP(7) = -.0058254
COSSLP(8) = -.0066783
COSSLP(9) = -.007513
COSSLP(10) = -.008327
COSSLP(11) = -.0091183
COSSLP(12) = -.0098845
COSSLP(13) = -.010624
COSSLP(14) = -.011334
COSSLP(15) = -.012013
COSSLP(16) = -.012659
COSSLP(17) = -.01327
COSSLP(18) = -.013845
COSSLP(19) = -.014382
COSSLP(20) = -.01488
COSSLP(21) = -.015337
COSSLP(22) = -.015751
COSSLP(23) = -.016123
COSSLP(24) = -.01645
COSSLP(25) = -.016733
COSSLP(26) = -.016969
COSSLP(27) = -.017159
COSSLP(28) = -.017302
COSSLP(29) = -.017398
COSSLP(30) = -.017445
COSSLP(31) = 1!

ISNX(1) = 0!
ISNX(2) = .0349
ISNX(3) = .06976
ISNX(4) = .10453
ISNX(5) = .13917
ISNX(6) = .17365
ISNX(7) = .20791
ISNX(8) = .24192
ISNX(9) = .27564
ISNX(10) = .30902
ISNX(11) = .34202
ISNX(12) = .37461
ISNX(13) = 1E+30

ISNY(1) = 0!
ISNY(2) = 2!
ISNY(3) = 4!
ISNY(4) = 6!
ISNY(5) = 8!
ISNY(6) = 10!
ISNY(7) = 12!
ISNY(8) = 14!
ISNY(9) = 16!
ISNY(10) = 18!
ISNY(11) = 20!
ISNY(12) = 22!
ISNY(13) = 1E+30

ISNSLP(1) = 57.307
ISNSLP(2) = 57.378
ISNSLP(3) = 57.517
ISNSLP(4) = 57.729
ISNSLP(5) = 58.013
ISNSLP(6) = 58.371
ISNSLP(7) = 58.805
ISNSLP(8) = 59.32
ISNSLP(9) = 59.917
ISNSLP(10) = 60.6
ISNSLP(11) = 61.375
ISNSLP(12) = 62.247
ISNSLP(13) = 1!
```

```
'DOUBLE SAFETY INTERLOCK VALUE
K6A = .0019531#
KRAD = .017432

'LIMITS
L10 = .02
L20 = .5
L30 = .24
L50 = 0!
L61 = -51.5745
L62 = 84.9
L100 = 15!
L110A = 275!
L110 = -95!
JJ = 0
KK = 0

LOCATE 3, 27: PRINT "PRESENT VALUE       MAX          MIN"
LOCATE 4, 2: PRINT "STRUT ANG, DEG:"
LOCATE 4, 45: PRINT "+19.0        -11.0"
LOCATE 5, 2: PRINT "TSWT POS, DEG:"
LOCATE 5, 46: PRINT "+1.0         -0.5"
LOCATE 6, 2: PRINT "TSWB POS, DEG:"
LOCATE 6, 46: PRINT "+1.0         -0.5"
LOCATE 7, 2: PRINT "MSWTF POS, DEG:"
LOCATE 7, 46: PRINT "+0.0         -4.5"
LOCATE 8, 2: PRINT "MSWTN POS, DEG:"
LOCATE 8, 46: PRINT "+0.0         -4.5"
LOCATE 9, 2: PRINT "MSWBF POS, DEG:"
LOCATE 9, 46: PRINT "+0.0         -4.5"
LOCATE 10, 2: PRINT "MSWBN POS, DEG"
LOCATE 10, 46: PRINT "+0.0         -4.5"
'OMITTED SIDE RENTRY FLAP ZERO TEST

LOCATE 11, 2: PRINT "REFTF POS, DEG:"
LOCATE 11, 45: PRINT "+15.0         0.0"
LOCATE 12, 2: PRINT "REFTN POS, DEG:"
LOCATE 12, 45: PRINT "+15.0         0.0"
LOCATE 13, 2: PRINT "REFBF POS, DEG:"
LOCATE 13, 45: PRINT "+15.0         0.0"
LOCATE 14, 2: PRINT "REFBN POS, DEG:"
LOCATE 14, 45: PRINT "+15.0         0.0"
LOCATE 15, 2: PRINT "PSTAT(PT2647), PSIA:"
LOCATE 15, 45: PRINT "+57.9        +14.7"
LOCATE 16, 2: PRINT "PTOT(PT2615), PSIA:"
LOCATE 16, 44: PRINT "+135.0        +14.7"
LOCATE 17, 2: PRINT "ROLL ANGLE, DEG:"
LOCATE 17, 44: PRINT "+275.0        -95.0"
LOCATE 18, 2: PRINT "MACH NO.:"
LOCATE 18, 47: PRINT "1.2          0.0"
LOCATE 19, 2: PRINT "TUNNEL Q, PSI:"
LOCATE 19, 44: PRINT " +53.0         0.0"

1 'CONTINUE
'SETUP FOR A-TO-D BOARD "A" (DT2801/5716A). THESE ARE THE "A" VOLTAGES!
BASE.ADDRESS = &H2DC
COMMAND.REGISTER = BASE.ADDRESS + 1
STATUS.REGISTER = BASE.ADDRESS + 1
DATA.REGISTER = BASE.ADDRESS
GOTO 1000
```

```
6 'SETUP FOR A-TO-D BOARD "B" (DT2801/5716A). THESE ARE THE "B" VOLTAGES!
BASE.ADDRESS = &H2EC
COMMAND.REGISTER = BASE.ADDRESS + 1
STATUS.REGISTER = BASE.ADDRESS + 1
DATA.REGISTER = BASE.ADDRESS
GOTO 2000

10 'SETUP FOR A-TO-D BOARD "C" (DT2801/5716A). THESE ARE THE "C" VOLTAGES!
BASE.ADDRESS = &H2F4
COMMAND.REGISTER = BASE.ADDRESS + 1
STATUS.REGISTER = BASE.ADDRESS + 1
DATA.REGISTER = BASE.ADDRESS
GOTO 3000

15 'CONVERSION TO ENGINEERING UNITS
'BOARD "A", A(1) THRU A(8), &H2DC (BASE.ADDRESS)
'A(1) IS GUIDE VANE ANGLE--ZT-2012
'A(2) IS STRUT POSITION--ZT-1603A
'A(3) IS PUMP SWASH PLATE--ZT-1602
'A(4) IS TSW TOP POSITION--ZT-1821
'A(5) IS TSW BOTTOM POSITION--ZT-1861
'A(6) IS MSW TOP FS POSITION--ZT-1501
'A(7) IS MSW TOP NS POSITION--ZT-1401
'A(8) IS MSW BOTTOM FS POSITION--ZT-1521

'BOARD "B", B(1) THRU B(8), &H2EC (BASE.ADDRESS)
'B(1) IS MSW BOTTOM NS POSITION--ZT-1421
'B(2) IS REF TOP FS POSITION--ZT-1541
'B(3) IS REF TOP NS POSITION--ZT-1441
'B(4) IS REF SIDE FS POSITION--ZT-1581
'B(5) IS REF SIDE NS POSITION--ZT-1481
'B(6) IS REF BOTTOM FS POSITION--ZT-1561
'B(7) IS REF BOTTOM NS POSITION--ZT-1461
'B(8) IS STRUT LIMIT ARMED (DISABE)--HS-4603

'BOARD "C", C(1) THRU C(8), &H2F4 (BASE.ADDRESS)
'C(1) IS PRESSURE (STATIC SIGNAL)--PT-2647
'C(2) IS TOTAL PRESSURE PO--PT-2615
'C(3) IS PRESSURE (STATIC RANGE CODE)--PT-2647

'!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

'CONVERT THE A,B, & C VOLATGE VALUES TO REAL NUMBERS
XGVAX = A(1) * (6.13933) + .18551
XSTRUT = A(2) * (-3.3113) + 19.4982
LOCATE 4, 28: PRINT USING "(###.### )"; XSTRUT
XPSP = A(3) * 204.7
ALPHAT = A(4) * (.165) - .6
LOCATE 5, 28: PRINT USING "(###.### )"; ALPHAT
ALPHAB = A(5) * (.165) - .6
LOCATE 6, 28: PRINT USING "(###.### )"; ALPHAB
BETATF = A(6) * (-.49) + .2
LOCATE 7, 28: PRINT USING "( ##.### )"; BETATF
BETATN = A(7) * (-.49) + .2
LOCATE 8, 28: PRINT USING "( ##.### )"; BETATN
BETABF = A(8) * (-.49) + .2
LOCATE 9, 28: PRINT USING "( ##.### )"; BETABF
BETABN = B(1) * (-.49) + .2
LOCATE 10, 28: PRINT USING "( ##.### )"; BETABN
```

```
GAMATF = B(2) * (1.57) - .2
LOCATE 11, 28: PRINT USING "( ##.### )"; GAMATF
GAMATN = B(3) * (1.57) - .2
LOCATE 12, 28: PRINT USING "( ##.### )"; GAMATN

'THESE ARE COMMENTED OUT SINCE THEY ARE NOT USED!!!!
'GAMAFS = B(4) * (1.57) - .2
'GAMANS = B(5) * (1.57) - .2

'PRINT USING "GAMAFS IS ####.###"; GAMAFS
'PRINT USING "GAMANS IS ####.###"; GAMANS
GAMABF = B(6) * (1.57) - .2
LOCATE 13, 28: PRINT USING "( ##.### )"; GAMABF
GAMABN = B(7) * (1.57) - .2
LOCATE 14, 28: PRINT USING "( ##.### )"; GAMABN
DISABE = B(8)

'RANGE SELECT VOLTAGE FOR PT-2647 (STATIC PRESSURE)
OLDRG1 = RANGE1
RANGE1 = C(3)
CHG1 = RANGE1 - OLDRG1
SKIP1 = TRUE

IF (ABS(CHG1) < .1) THEN
   SKIP1 = FALSE
   END IF

'IF AUTORANGEING IN PROGRESS, SKIP PRESSURE UPDATE FOR 3 SECONDS
IF (SKIP1) THEN
   CNT1 = 1
   END IF

'RESET COUNTER ON POWERUP AND DELAY (2.5 SEC'S)
IF ((CNT1 > 0) AND (CNT1 < 25)) THEN
   CNT1 = CNT1 + 1
   ELSEIF ((CNT1 >= 25) OR (CNT1 < 0)) THEN
   CNT1 = 0
   END IF
IF (CNT1 <> 0) GOTO 100

'DETERMINE RANGE FOR PSTAT
IF RANGE1 <= 3.5 THEN
   PSTAT = C(1) * 1.5
   ELSEIF ((RANGE1 > 3.5) AND (RANGE1 <= 4!)) THEN
   PSTAT = C(1) * 4.5
   ELSEIF ((RANGE1 > 4)) THEN
   PSTAT = C(1) * 15!
   END IF
LOCATE 15, 28: PRINT USING "(###.### )"; PSTAT
100 'CONTINUE
'TOTAL PRESSURE FROM PT-2615
XPRES = C(2) * 29.963 - .134
LOCATE 16, 28: PRINT USING "(###.### )"; XPRES

GOTO 101

1000 'ANALOG TO DIGITAL CONVERSION SUBROUTINE.  BOARD A--"A'S".
'CHECK FOR LEGAL STATUS REGISTER
STATUS = INP(STATUS.REGISTER)
IF NOT ((STATUS AND &H70) = 0) THEN
```

```
   GOTO 5000
   ELSE
   GOTO 5001
   END IF

5001 'STOP AND CLEAR THE DT-2801/5716A BOARD
OUT COMMAND.REGISTER, CSTOP
TEMP = INP(DATA.REGISTER)
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT

WAIT STATUS.REGISTER, COMMAND.WAIT
OUT COMMAND.REGISTER, CCLEAR
FOR I = 1 TO 8
CHANNEL = I - 1
'WRITE READ A/D IMMEDIATE COMMAND
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT

WAIT STATUS.REGISTER, COMMAND.WAIT
OUT COMMAND.REGISTER, CADIN
'WRITE A/D GAIN BYTE
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
OUT DATA.REGISTER, GAIN.CODE
'WRITE A/D CHANNEL BYTE
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
OUT DATA.REGISTER, CHANNEL
'READ TWO BYTES OF A/D DATA FROM THE DATA OUT REGISTER AND
'COMBINE THE TWO BYTES INTO ONE WORD.
WAIT STATUS.REGISTER, READ.WAIT
LOW = INP(DATA.REGISTER)
WAIT STATUS.REGISTER, READ.WAIT
HIGH = INP(DATA.REGISTER)
DATA.VALUE# = HIGH * 256 + LOW
'LOCATE 21, 60: PRINT USING "#######.#"; DATA.VALUE#
IF DATA.VALUE# > 32767 THEN
   DATA.VALUE# = DATA.VALUE# - 65536
   ELSE
   GOTO 5002
   END IF

5002 'CONTINUE

3 ' CALCULATE THE A/D READING IN VOLTS
'VOLTS#=(RANGE*DATA.VALUE#/FACTOR#/GAIN)
VOLTS# = SCALE * DATA.VALUE#
A(I) = VOLTS#
'LOCATE 1, 1: PRINT USING "###.###"; A(1)
'LOCATE 1, 10: PRINT USING "###.###"; A(2)
'LOCATE 1, 20: PRINT USING "###.###"; A(3)
'LOCATE 1, 30: PRINT USING "###.###"; A(4)
'LOCATE 1, 40: PRINT USING "###.###"; A(5)
'LOCATE 1, 50: PRINT USING "###.###"; A(6)
'LOCATE 1, 60: PRINT USING "###.###"; A(7)
'LOCATE 1, 70: PRINT USING "###.###"; A(8)

IF (I = 8) THEN
   GOTO 5
   ELSE
   GOTO 4
   END IF
4 NEXT I
```

```
5 GOTO 6


2000 'ANALOG TO DIGITAL CONVERSION SUBROUTINE.  BOARD B--"B'S".
'CHECK FOR LEGAL STATUS REGISTER
STATUS = INP(STATUS.REGISTER)
IF NOT ((STATUS AND &H70) = 0) THEN
   GOTO 5000
   ELSE
   GOTO 7
   END IF
7 'STOP AND CLEAR THE DT-2801/5716A BOARD
OUT COMMAND.REGISTER, CSTOP
TEMP = INP(DATA.REGISTER)
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
WAIT STATUS.REGISTER, COMMAND.WAIT
OUT COMMAND.REGISTER, CCLEAR
FOR I = 1 TO 8
CHANNEL = I - 1
'WRITE READ A/D IMMEDIATE COMMAND
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
WAIT STATUS.REGISTER, COMMAND.WAIT
OUT COMMAND.REGISTER, CADIN
'WRITE A/D GAIN BYTE
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
OUT DATA.REGISTER, GAIN.CODE
'WRITE A/D CHANNEL BYTE
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
OUT DATA.REGISTER, CHANNEL

'READ TWO BYTES OF A/D DATA FROM THE DATA OUT REGISTER, AND
'COMBINE THE TWO BYTES INTO ONE WORD
WAIT STATUS.REGISTER, READ.WAIT
LOW = INP(DATA.REGISTER)
WAIT STATUS.REGISTER, READ.WAIT
HIGH = INP(DATA.REGISTER)
DATA.VALUE# = HIGH * 256 + LOW
IF DATA.VALUE# > 32767 THEN
   DATA.VALUE# = DATA.VALUE# - 65536
   ELSE
   GOTO 8
   END IF

8 'CONTINUE

9     'CALCULATE THE A/D READINGS IN VOLTS
'VOLTS# = (RANGE * DATA.VALUE# / FACTOR# / GAIN)
VOLTS# = SCALE * DATA.VALUE#
B(I) = VOLTS#
'LOCATE 2, 1: PRINT USING "###.###"; B(1)
'LOCATE 2, 10: PRINT USING "###.###"; B(2)
'LOCATE 2, 20: PRINT USING "###.###"; B(3)
'LOCATE 2, 30: PRINT USING "###.###"; B(4)
'LOCATE 2, 40: PRINT USING "###.###"; B(5)
'LOCATE 2, 50: PRINT USING "###.###"; B(6)
'LOCATE 2, 60: PRINT USING "###.###"; B(7)
'LOCATE 2, 70: PRINT USING "###.###"; B(8)

IF (I = 8) THEN
```

```
   GOTO 10
   ELSE
   GOTO 11
   END IF
11 NEXT I

3000 'ANALOG TO DIGITAL CONVERSION SUBROUTINE.  BOARD C--"C'S".
'CHECK FOR LEGAL STATUS REGISTER
STATUS = INP(STATUS.REGISTER)
IF NOT ((STATUS AND &H70) = 0) THEN
   GOTO 5000
   ELSE
   GOTO 12
   END IF
12  'STOP AND CLEAR THE DT-2801/5716A BOARD
OUT COMMAND.REGISTER, CSTOP
TEMP = INP(DATA.REGISTER)
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
WAIT STATUS.REGISTER, COMMAND.WAIT
OUT COMMAND.REGISTER, CCLEAR
FOR I = 1 TO 3
CHANNEL = I - 1

'WRITE READ A/D IMMEDIATE COMMAND
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
WAIT STATUS.REGISTER, COMMAND.WAIT
OUT COMMAND.REGISTER, CADIN
'WRITE A/D GAIN BYTE
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
OUT DATA.REGISTER, GAIN.CODE
'WRITE A/D CHANNEL BYTE
WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
OUT DATA.REGISTER, CHANNEL

'READ TWO BYTES OF A/D DATA FROM THE DATA OUT REGISTER, AND COMBINE
'THE TWO BYTES INTO ONE WORD
WAIT STATUS.REGISTER, READ.WAIT
LOW = INP(DATA.REGISTER)
WAIT STATUS.REGISTER, READ.WAIT
HIGH = INP(DATA.REGISTER)
DATA.VALUE# = HIGH * 256 + LOW
   IF DATA.VALUE# > 32767 THEN
   DATA.VALUE# = DATA.VALUE# - 65536
   ELSE
   GOTO 13
   END IF

13 'CONTINUE

14 'CALCULATE THE A/D READINGS IN VOLTS
'VOLTS#=(RANGE*DATA.VALUE#/FACTOR#/GAIN)
VOLTS# = SCALE * DATA.VALUE#
C(I) = VOLTS#
'LOCATE 3, 1: PRINT USING "###.##"; C(1)
'LOCATE 3, 10: PRINT USING "###.##"; C(2)
'LOCATE 3, 20: PRINT USING "###.##"; C(3)
IF (I = 3) THEN
   GOTO 15
   ELSE
   GOTO 16
   END IF
16 NEXT I

5000 'ILLEGAL STATUS REGISTER
PRINT
PRINT "FATAL ERROR-ILLEGAL STATUS REGISTER VALUE!"
PRINT "STATUS REGISTER VALUE IS"; HEX$(STATUS); "HEXIDECIMAL"
PRINT : BEEP: BEEP
END
'NEED TO RESET WATCHDOG TO KEEP FROM TIMING OUT.  THIS IS NOT USED IN
'THIS SIMULATION, BUT INSERTED TO KEEP IN PARALLEL WITH OLD MICRO
'SUBROUTINE "RELAYS". WILL BE ADDED LATER, BUT FOR NOW WILL PRINT TO
'THE SCREEN FOR RELAYS.
'CALL RELAYS

101 'CONTINUE
'DIAL5 SUBROUTINE---THIS READS IN THE SETTING OF FIVE DIFFERENT THUMBWHEEL
'SWITCHES AND CONVERTS THE SETTINGS TO REAL NUMBER VALUES USING THE
'DT-2817 BOARDS WITH FOUR DIGITAL I/O PORTS PER BOARD

'DEFINE THE DIGITAL I/O CONSTANTS---SEE "DT-2817 USER MANUAL"
'CONTROL REGISTER WITH FACTORY BASE ADDRESS
CONTROL.REGISTER% = &H228
'INPUT PORT ADDRESSES
DATA.PORT0 = &H229
DATA.PORT1 = &H22A
DATA.PORT2 = &H22B
DATA.PORT3 = &H22C

'ALL PORTS SET FOR INPUTS (&H0=0000)
OUT CONTROL.REGISTER%, &H0
'READ THE INPUT PORTS
PORT.VALUE.READ0 = INP(DATA.PORT0)
PORT.VALUE.READ1 = INP(DATA.PORT1)
PORT.VALUE.READ2 = INP(DATA.PORT2)
PORT.VALUE.READ3 = INP(DATA.PORT3)
DATA.VALUE(1) = PORT.VALUE.READ0 / 10
DATA.VALUE(2) = PORT.VALUE.READ1
DATA.VALUE(3) = DATA.VALUE(1) + DATA.VALUE(2)
DATA.VALUE(4) = PORT.VALUE.READ2
DATA.VALUE(5) = PORT.VALUE.READ3
'THIS IS THE SIMULATED DIALED-IN THUMBWHEEL VALUE FOR XQDIAL!
XQDIAL = DATA.VALUE(3)
'LOCATE 20, 27: PRINT USING "####.####"; XQDIAL
'THIS IS THE SIMULATED DIALED-IN THUMBWHEEL VALUES FOR THE
'POSITIVE AND NEGATIVE PITCH LIMITS---XSTRDP & XSTRDM
XSTRDP = DATA.VALUE(4)
XSTRDM = (-1) * DATA.VALUE(5)
'LOCATE 21, 54: PRINT USING "+###.#"; XSTRDP
'LOCATE 20, 55: PRINT USING "###.#"; XSTRDM

GOTO 6001

6001 'CONTINUE
XROLDP = 275!
XROLDM = -95!
'PRINT USING "XROLDM IS ###.##"; XROLDM
'PRINT USING "XROLDP IS ###.##"; XROLDP

'ENCDIN SUBROUTINE---THIS READS IN THE SETTINGS OF THE MODEL ROLL
```

```
'ANGLE (IN A FIVE CHARACTOR BCD CODE FROM DIO PORTS).  FOR NOW SET
'ROL TO A VALUE.
'ROL = D(2) * 10000 + D(3) * 1000 + D(4) * 100 + D(5) * 10 + D(6)
'ANGLE = (ROL / 100) - 35.4
ANGLE = 0!
LOCATE 17, 28: PRINT USING "(###.### )"; ANGLE

'CALCD SUBROUTINE(GAMATF, CDTF, IC(1))---THIS CALCULATES THE RADIUS ARM "CD" (LE
'FLAP TIP TO MODEL SUPPORT WAL PIVOT FOR TOP AND BOTTOM REF'S) AS A FUNCTION
'OF THE RENTRY FLAP ANGLE, GAMA FOR TOP AND BOTTOM REF'S.
DAT = GAMATF
   'BEGIN INDEX SUB
    J = 15 'THIS IS IC(1)
    IF DAT >= COSX(J) THEN
2001 'CONTINUE
      IF DAT >= COSX(J + 1) THEN
      J = J + 1
      GOTO 2001
      END IF
    ELSE
3001 'CONTINUE
      IF J = 1 THEN GOTO 201
      J = J - 1
      IF DAT < COSX(J) THEN GOTO 3001
    END IF
    'END INDEX SUB
201 INTCOS = COSY(J) + (GAMATF - COSX(J)) * COSSLP(J)
CDTF = SQR(KCD1 + KCD2 * INTCOS)

'CALCD SUBROUTINE (GAMATN,CDTN,IC(1))
DAT = GAMATN
   'BEGIN INDEX SUB
    J = 15 'THIS IS IC(1)
    IF DAT >= COSX(J) THEN
202  'CONTINUE
      IF DAT >= COSX(J + 1) THEN
         J = J + 1
         GOTO 202
      END IF
   ELSE
203  'CONTINUE
      IF J = 1 THEN GOTO 204
      J = J - 1
      IF DAT < COSX(J) THEN GOTO 203
   END IF
   'END INDEX SUB
204 INTCOS = COSY(J) + (GAMATN - COSX(J)) * COSSLP(J)
CDTN = SQR(KCD1 + KCD2 * INTCOS)

'CALCD SUBROUTINE (GAMABF, CDBF, IC(2))
DAT = GAMABF
   'BEGIN INDEX SUB
    J = 15'THIS IS IC(2)
    IF DAT >= COSX(J) THEN
4001 'CONTINUE
   IF DAT >= COSX(J + 1) THEN
        J = J + 1
        GOTO 4001
      END IF
   ELSE
```

```
5003 'CONTINUE
   IF J = 1 THEN GOTO 303
   J = J - 1
   IF DAT < COSX(J) GOTO 5003
   END IF
   'END INDEX SUB
303 INTCOS = COSY(J) + (GAMABF - COSX(J)) * COSSLP(J)
CDBF = SQR(KCD1 + KCD2 * INTCOS)

'CALCD SUBROUTINE  (GAMABN, CDBN, IC(2))
DAT = GAMABN
   'BEGIN INDEX SUB
    J = 15 'THIS IS IC(2)
    IF DAT >= COSX(J) THEN
304  'CONTINUE
     IF DAT >= COSX(J + 1) THEN
     J = J + 1
     GOTO 304
     END IF
   ELSE
305  'CONTINUE
     IF J = 1 THEN GOTO 306
     J = J - 1
     IF DAT < COSX(J) THEN GOTO 305
    END IF
    'END INDEX SUB
306  INTCOS = COSY(J) + (GAMABN - COSX(J)) * COSSLP(J)
CDBN = SQR(KCD1 + KCD2 * INTCOS)

'CALN SUBROUTINE (GAMATF, NTF, XNTF, CDTF, IC(3), IS(1))
'THIS CALCULATES THE ANGLE "NU" FORMED BY THE MODEL
'SUPPORT WALL AND THE RADIUS ARMS CDTF, CDTN, CDBF, AND CDBN
DAT = 90! - GAMATF
     'BEGIN INDEX SUBROUTINE
     J = 15 'THIS IS IC(3)
     IF DAT >= COSX(J) THEN
8000 'CONTINUE
          IF DAT >= COSX(J + 1) THEN
             J = J + 1
             GOTO 8000
          END IF
    ELSE
8002 'CONTINUE
          IF J = 1 THEN GOTO 8001
          J = J - 1
          IF DAT < COSX(J) THEN GOTO 8002
     END IF
     'END INDEX SUBROUTINE
8001  INTCOS = COSY(J) + (DAT - COSX(J)) * COSSLP(J)
DAT = 60! * INTCOS / CDTF
     'BEGIN INDEX SUBROUTINE
     J = 5 'THIS IS IS(1)
     IF DAT >= ISNX(J) THEN
8003 'CONTINUE
          IF DAT >= ISNX(J + 1) THEN
          J = J + 1
          GOTO 8003
          END IF
    ELSE
8005 'CONTINUE
```

```
          IF J = 1 THEN GOTO 8004
          J = J - 1
          IF DAT < ISNX(J) THEN GOTO 8005
     END IF
     'END INDEX SUBROUTINE
8004  NTF = ISNY(J) + (DAT - ISNX(J)) * ISNSLP(J)
XNTF = NTF * KRAD

'CALN SUBROUTINE (GAMATN, NTN, XNTN, CDTN, IC(3), IS(1))
DAT = 90! - GAMATN
     'BEGIN INDEX SUB
     J = 15 'THIS IS IC(3)
     IF DAT >= COSX(J) THEN
8006 'CONTINUE
          IF DAT >= COSX(J + 1) THEN
          J = J + 1
          GOTO 8006
          END IF
    ELSE
8007 'CONTINUE
          IF J = 1 THEN GOTO 8008
          J = J - 1
          IF DAT < COSX(J) THEN GOTO 8007
     END IF
     'END INDEX SUB
8008 INTCOS = COSY(J) + (DAT - COSX(J)) * COSSLP(J)
DAT = 60! * INTCOS / CDTN
     'BEGIN INDEX SUB
     J = 5 'THIS IS IS(1)
     IF DAT >= ISNX(J) THEN
8009 'CONTINUE
          IF DAT >= ISNX(J + 1) THEN
          J = J + 1
          GOTO 8009
          END IF
    ELSE
8010 'CONTINUE
          IF J = 1 THEN GOTO 8011
          J = J - 1
          IF DAT < ISNX(J) THEN GOTO 8010
     END IF
     'END INDEX SUB
8011 NTN = ISNY(J) + (DAT - ISNX(J)) * ISNSLP(J)
XNTN = NTN * KRAD

'CALN SUBROUTINE (GAMABF, NBF, XNBF, CDBF, IC(5), IS(3))
DAT = 90! - GAMABF
     'BEGIN INDEX SUB
     J = 15 'THIS IS IC(5)
     IF DAT >= COSX(J) THEN
8012 'CONTINUE
          IF DAT >= COSX(J + 1) THEN
          J = J + 1
          GOTO 8012
          END IF
      ELSE
8013 'CONTINUE
          IF J = 1 THEN GOTO 8014
          J = J - 1
          IF DAT < COSX(J) THEN GOTO 8013
          END IF
          'END INDEX SUB
8014 INTCOS = COSY(J) + (DAT - COSX(J)) * COSSLP(J)
DAT = 60! * INTCOS / CDBF
     'BEGIN INDEX SUB
     J = 5 'THIS IS IS(3)
     IF DAT >= ISNX(J) THEN
8015 'CONTINUE
          IF DAT >= ISNX(J + 1) THEN
          J = J + 1
          GOTO 8015
          END IF
     ELSE
8016 'CONTINUE
          IF J = 1 THEN GOTO 8017
          J = J - 1
          IF DAT < ISNX(J) THEN GOTO 8016
     END IF
     'END INDEX SUB
8017 NBF = ISNY(J) + (DAT - ISNX(J)) * ISNSLP(J)
XNBF = NBF * KRAD

'CALN SUBROUTINE (GAMABN, NBN, XNBN, CDBN, IC(5), IS(3))
DAT = 90! - GAMABN
     'BEGIN INDEX SUB
     J = 15 'THIS IS IC(5)
     IF DAT >= COSX(J) THEN
8018 'CONTINUE
          IF DAT >= COSX(J + 1) THEN
          J = J + 1
          GOTO 8018
          END IF
    ELSE
8019 'CONTINUE
          IF J = 1 THEN GOTO 8020
          J = J - 1
          IF DAT < COSX(J) THEN GOTO 8019
    END IF
    'END INDEX SUB
8020 INTCOS = COSY(J) + (DAT - COSX(J)) * COSSLP(J)
DAT = 60! * INTCOS / CDBN
     'BEGIN INDEX SUB
     J = 5 'THIS IS IS(3)
     IF DAT >= ISNX(J) THEN
8021 'CONTINUE
          IF DAT >= ISNX(J + 1) THEN
          J = J + 1
          GOTO 8021
          END IF
    ELSE
8022 'CONTINUE
          IF J = 1 THEN GOTO 8023
          J = J - 1
          IF DAT < ISNX(J) THEN GOTO 8022
    END IF
    'END INDEX SUB
8023 NBN = ISNY(J) + (DAT - ISNX(J)) * ISNSLP(J)
XNBN = NBN * KRAD

'MACHIN SUBROUTINE---CALCULATE THE TUNNEL MACH NUMBER
```

```
PRATO = XPRES / PSTAT
IF (PRATO < 1!) THEN
   GOTO 24
   ELSE
   GOTO 25
   END IF
24 PRATO = 1!
25 MACHSQ = (5! * (PRATO ^ .28571 - 1!))

'PRINT USING "PRATO IS ####.###"; PRATO
MACH = SQR(MACHSQ)
LOCATE 18, 28: PRINT USING "( ##.### )"; MACH

'CALQ SUBROUTINE---CALCULATE THE OPERATIONAL TUNNEL "Q" FROM THE
'COMPUTED MACH NUMBER AND THE MEASURED PRESSURE
QCOMP = XPRES * (KQ1 * MACHSQ / (1! + .2 * MACHSQ) ^ 3.5)
LOCATE 19, 28: PRINT USING "(###.### )"; QCOMP

'PERFORM THE INTERLOCK TESTS

'TEST 01---THIS ROUTINE TESTS TO ENSURE THAT THE TOP AND BOTTOM
'TEST SECTION WALLS ARE BOTH WITHIN +,- 0.02 DEGREES OF THEIR ZERO
'POSITION BEFORE "PERMISSION" IS GRANTED TO INSERT OR REMOVE
'THE MODEL ACCESS HOUSINGS. NO INTERFERENCE ALLOWED. IF BOTH TOP
'AND BOTTOM WALLS ARE WITHIN LIMIT, THE "PERMISSIVE" SEQUENCER RELAY
'IS DRIVEN.

'COMPARE THE TSW TOP AND BOTTOM ANGLES WITH THEIR LIMIT AND SET
'FLAG FOR SEQUENCER OUTPUT TRUE IF ALPHA (TSW) TOP  OR BOT
'IS >= +0.02 OR <= -0.02.  L10 = 0.02
IF (ALPHAT >= L10) THEN
   FLAG25(3) = 0
   GOTO 26
   ELSE
   FLAG25(3) = 1
   GOTO 555
   END IF
26 COLOR 12
LOCATE 5, 45: PRINT "TSWT>= PLUS ZERO LIM"
LL = 1
GOTO 27
555 COLOR 14
IF (LL = 1) THEN
   GOTO 271
   ELSE
   GOTO 27
   END IF
271 LOCATE 5, 45: PRINT "                                        "
    LL = 0
LOCATE 5, 46: PRINT "+1.0        -0.5"
27 IF (ALPHAT <= -L10) THEN
   FLAG25(3) = 0
   GOTO 28
   ELSE
   FLAG25(3) = 1
   GOTO 301
   END IF
28 COLOR 12
LOCATE 5, 45: PRINT "TSWT<= NEG ZERO LIM"
LLL = 1
```

```
GOTO 29
301 COLOR 14
IF (LLL = 1) THEN
   GOTO 302
   ELSE
   GOTO 29
   END IF
302 LOCATE 5, 45: PRINT "                              "
LLL = 0
LOCATE 5, 46: PRINT "+1.0        -0.5"
29 IF (ALPHAB >= L10) THEN
   FLAG25(3) = 0
   GOTO 30
   ELSE
   FLAG25(3) = 1
   GOTO 311
   END IF
30 COLOR 12
LOCATE 6, 45: PRINT "TSWB>= PLUS ZERO LIM"
LLLL = 1
GOTO 32
311 COLOR 14
   IF (LLLL = 1) THEN
   GOTO 312
   ELSE
   GOTO 32
   END IF
312 LOCATE 6, 45: PRINT "                            "
   LLLL = 0
LOCATE 6, 46: PRINT "+1.0        -0.5"
GOTO 32

32 IF (ALPHAB <= -L10) THEN
   FLAG25(3) = 0
   GOTO 361
   ELSE
   FLAG25(3) = 1
   GOTO 362
   END IF
361 COLOR 12
LOCATE 6, 45: PRINT "TSWB<= NEG ZERO LIM"
LLLLL = 1
GOTO 364
362 COLOR 14
   IF (LLLLL = 1) THEN
   GOTO 363
   ELSE
   GOTO 364
   END IF
363 LOCATE 6, 45: PRINT "                           "
LLLLL = 0
LOCATE 6, 46: PRINT "+1.0        -0.5"
364 'CONTINUE

'TEST 02 SUBROUTINE (ALPHAT, BETATF, XNTF, CDTF, FLAG24(1))
'THIS ROUTINE TESTS THE DISTANCES BETWEEN DIFFERENT
'WALL SECTIONS TO ENSURE AGAINST WALLS COLLIDING.  THE TEST HAS TWO
'SEPARATE PARTS WHICH MUST BE SATISFIED.
'TEST 2.1---ENSURES THAT THE DISTANCE BETWEEN THE TSW AND MSW IS > = 0.5
'INCHES.
```

```
'TEST 2.2---ENSURES THAT THE DISTANCE BETWEEN THE TSW AND REF IS >= 0.5
'INCHES.  IF EITHER PART IS NOT SATISFIED, PRINT A MESSAGE.

FLAG24(1) = 0
X = BETATF * KRAD
Y = ALPHAT * KRAD

'TEST SECTION WALL TOP-MODEL SUPPORT WALL TOP FAR SIDE. L20=0.5!
TEST2(1) = KCL + KLENBD * X - (KLENA * Y)
TEST2(2) = KCL + CDTF * (X + XNTF) - (KSTA36 - CDTF) * Y

'LOCATE 20, 2: PRINT USING "(####.##)"; TEST2(1)
'LOCATE 20, 12: PRINT USING "(####.##)"; TEST2(2)
'IF THE TEST VALUE IS LESS THAN ITS LIMIT PRINT TO SCREEN (SET A FLAG LATER)
IF ((TEST2(1) <= L20) OR (TEST2(2) <= L20)) THEN
   FLAG24(1) = 1
   COLOR 12
   LOCATE 21, 2: PRINT "TSWT/MSWTF/REFTF COLLIDE"
   GOTO 368
   ELSE
   GOTO 369
369 COLOR 14
   LOCATE 21, 2: PRINT "                         "
   END IF
368 'CONTINUE

'TEST 02 SUBROUTINE (ALPHAT, BETATN, XNTN, CDTN, FLAG24(1))
FLAG24(1) = 0
X = BETATN * KRAD
Y = ALPHAT * KRAD
TEST2(1) = KCL + KLENBD * X - (KLENA * Y)
TEST2(2) = KCL + CDTN * (X + XNTN) - (KSTA36 - CDTN) * Y
'LOCATE 20, 2: PRINT USING "(####.##)"; TEST2(1)
'LOCATE 20, 12: PRINT USING "(####.##)"; TEST2(2)
'IF THE TEST VALUE IS LESS THAN ITS LIMIT PRINT TO SCREEN
IF ((TEST2(1) <= L20) OR (TEST2(2) <= L20)) THEN
   FLAG24(1) = 1
   COLOR 12
   LOCATE 22, 2: PRINT "TSWT/MSWTN/REFTN COLLIDE"
   GOTO 366
   ELSE
   GOTO 367
   END IF
367 COLOR 14
   LOCATE 22, 2: PRINT "                         "
366 'CONTINUE

'TEST 02 SUBROUTINE (ALPHAB, BETABF, XNBF, CDBF, FLAG24(3))
FLAG24(3) = 0
X = BETABF * KRAD
Y = ALPHAB * KRAD

'TEST SECTION WALL BOTTOM- MODEL SUPPORT WALL BOTTOM FAR
TEST2(1) = KCL + KLENBD * X - (KLENA * Y)
TEST2(2) = KCL + CDBF * (X + XNBF) - (KSTA36 - CDBF) * Y
'LOCATE 20, 2: PRINT USING "(####.##)"; TEST2(1)
'LOCATE 20, 12: PRINT USING "(####.##)"; TEST2(2)

IF TEST2(1) <= L20 OR TEST2(2) <= L20 THEN
   FLAG24(3) = 1
   COLOR 12
   LOCATE 23, 2: PRINT "TSWB/MSWBF/REFBF COLLIDE"
   GOTO 371
   ELSE
   GOTO 372
   END IF
372 COLOR 14
LOCATE 23, 2: PRINT "                              "
371 'CONTINUE

'TEST 02 SUBROUTINE (ALPHAB, BETABN, XNBN, CDBN, FLAG24(3))
FLAG24(3) = 0
X = BETABN * KRAD
Y = ALPHAB * KRAD
TEST2(1) = KCL + KLENBD * X - (KLENA * Y)
TEST2(2) = KCL + CDBN * (X + XNBN) - (KSTA36 - CDBN) * Y
'LOCATE 20, 2: PRINT USING "(####.##)"; TEST2(1)
'LOCATE 20, 12: PRINT USING "(####.##)"; TEST2(2)
IF TEST2(1) <= L20 OR TEST2(2) <= L20 THEN
   FLAG24(3) = 1
   TMR = 1
   COLOR 12
   LOCATE 20, 25: PRINT "TSWB/MSWBN/REFBN COLLIDE"
   GOTO 373
   ELSE
   GOTO 374
   END IF
374 COLOR 14
LOCATE 20, 25: PRINT "                              "
   TMR = 0
373    'CONTINUE

'TEST 03 SUBROUTINE---THIS ROUTINE TESTS TO SEE IF
'       (1) NS AND FS BTM MSW ARE IN SYNC
'       (2) NS AND FS TOP MSW ARE IN SYNC
'CALCULATE THE ABSOLUTE DIFFERENCE IN NEAR AND FAR ANGLES--COMPARE
'THE DIFFERENCE WITH THE LIMIT AND SET THE FLAG TRUE IF THE LIMIT IS
'EXCEEDED. L30=0.24!
IF (ABS(BETABN - BETABF) >= L30) THEN
   BMSWS = 1
   ELSE
   BMSWS = 0
END IF

IF (ABS(BETATN - BETATF) >= L30) THEN
   TMSWS = 1
   ELSE
   TMSWS = 0
END IF
IF (BMSWS OR TMSWS) THEN
   FLAG24(4) = 1
   GOTO 44
   ELSE
   FLAG24(4) = 0
   GOTO 441
   END IF
44 COLOR 12
LOCATE 21, 31: PRINT "MSW NOT IN SYNC"
GOTO 45
441 COLOR 14
```

```
LOCATE 21, 31: PRINT "                    "
45 'CONTINUE

'TEST 05 SUBROUTINE---THIS ROUTINE TESTS TO ENSURE THAT THE ANGLE OF THE
'SIDE RENTRY FLAPS ARE NOT LESS THAN OR EQUAL TO 0.0 DEGREES.  IF EITHER
'OF THE SIDE RENTRY FLAP POSITIONS IS 0 OR NEGATIVE, A FLAG IS SET FOR
'OUTPUT TO THE SIDE WALL INTERFERENCE SEQUENCER INTERLOCK.
'CHECK FAR SIDE AND NEAR SIDE, SIDE REF ANGLES AND SET FLAG TRUE IF <= 0.0.
'L50 = 0.0!
IF (GAMAFS <= L50) THEN
   GOTO 46
   ELSE
   GOTO 47
   END IF
46  FLAG242 = TRUE
COLOR 12
'LOCATE 20, 25: PRINT "FS--SIDE REF IS 0.0 OR NEG"

47 'CONTINUE
IF (GAMANS <= L50) THEN
   GOTO 48
   ELSE
   GOTO 49
   END IF
48 FLAG242 = TRUE
COLOR 12
'LOCATE 21, 25: PRINT "NS--SIDE REF IS 0.0 OR NEG"

49 FLAG242 = FALSE
'CONTINUE

'TEST 06 SUBROUTINE---THIS ROUTINE TESTS THE DISTANCE BETWEEN
'THE STRUT AND THE MSW TO ENSURE THEY DO NOT COLLIDE.  THIS
'TEST HAS TWO PARTS--TEST 6.1 FOR THE TOP MSW CLEARANCE AND TEST
'6.2 FOR THE BOTTOM MSW.
'CHECK IF STRUT IS ABOVE OR BELOW LEVEL
'THE TOP MSW CLEARANCE AND TEST 6.2 FOR THE BOTTOM MSW.
'CHECK IF THE STRUT IS ABOVE OR BELOW LEVEL
FLAG16(3) = 0
FLAG16(4) = 0
IF (XSTRUT < 0!) THEN
   IF (BETATF <= BETATN) THEN
      BETA6 = BETATF
   ELSE
      BETA6 = BETATN
   END IF

   'COMPUTE THE STRUT VELOCITY (USING PSP)
    KS = K6A * XPSP
    KC = K117
    STRFLG = 1

ELSE

'TEST 6.2, STRUT-BOTTOM MSW COMPARE NEAR & FAR SIDES FOR LARGEST BETA.
'BETA IS ALWAYS NEGATIVE
   IF (BETABF <= BETABN) THEN
       BETA6 = BETABF
   ELSE
       BETA6 = BETABN
   END IF

   'CONTINUE--COMPUTE THE STRUT VELOCITY (USING PSP)
   KS = K6A * XPSP
   KC = -K40
   STRFLG = 0
END IF

'COMPUTE THE INTEFERENCE VALUE
STM = K45 * (XSTRUT - KS) + KC * BETA6

IF (STRFLG) THEN
   STMSWT = STM

'CHECK THE INTERFERENCE VALUE AGAINST THE NEGATIVE LIMIT (L61 = -51.5745)

   IF (STMSWT <= L61) THEN
      FLAG16(3) = 1
      COLOR 12
      LOCATE 22, 31: PRINT "STMSWT INTF'NCE "
      ELSE
      COLOR 14
      LOCATE 22, 31: PRINT "                 "
   END IF

ELSE
   STMSWB = STM

   'CHECK THE INTERFERENCE VALUE AGAINST THE POSITIVE LIMIT (L62 = +84.9)

   IF (STMSWB >= L62) THEN
      FLAG16(4) = 1
      COLOR 12
      LOCATE 23, 31: PRINT "STMSWB INTF'NCE"
      ELSE
      COLOR 14
      LOCATE 23, 31: PRINT "               "
   END IF
END IF

'TEST 07 SUBROUTINE--THIS ROUTINE TESTS TO INSURE THAT THE QCOMP OF THE
'TUNNEL DOES NOT EXCEED THE Q THUMBWHEEL VALUES OR THE STRUT POSITIVE
'OR NEGATIVE THUMBWHEEL VALUES.  ALSO, IT TESTS TUNNEL MACH NUMBER SQUARE
'AGAINST 1.5625 (I.E. M = 1.25).  IT ALSO CHECKS FOR SENSOR SIGNAL
'OUT-OF-RANGE FOR STRUT, PSTAT, PTOTAL, AND RANGE CODE (FOR PSTAT).

FLAG16(2) = 0
'IF (DISABE > 4!) THEN, IGNORE MODEL PITCH THUMBWHEEL LIMITS!!!
IF (DISABE > 4) THEN
   GOTO 65
   ELSE
   GOTO 651
   END IF
651 'CONTINUE
IF (XSTRUT >= XSTRDP OR XSTRUT <= XSTRDM) THEN
   FLAG16(2) = 1
   STRTMW = 1
   COLOR 12
  LOCATE 4, 45: PRINT "STRT THM'WL XEE'ED"
   GOTO 64
   ELSE
   GOTO 65
   END IF
65 COLOR 14
  LOCATE 4, 45: PRINT "+19.0       -11.0   "
   STRTMW = 0
64 'CONTINUE

IF (QCOMP >= XQDIAL) THEN
   FLAG16(2) = 1
   GOTO 66
   ELSE
   GOTO 671
   END IF
66 COLOR 12
LOCATE 19, 44: PRINT "TUNNEL Q EXCEEDED "
   TQ = 1
   GOTO 67
671 COLOR 14
IF (TQ = 1) THEN
   GOTO 672
   ELSE
   GOTO 67
   END IF
672 LOCATE 19, 44: PRINT "                     "
   TQ = 0
LOCATE 19, 44: PRINT " +53.0        0.0"
67 'CONTINUE

'LIMIT TUNNEL MACH NUMBER
IF (MACHSQ >= 1.5625) THEN
   FLAG16(2) = 1
   COLOR 12
   LOCATE 18, 41: PRINT "TUNNEL MACH NO. EXCEEDED"
   GOTO 68
   ELSE
   GOTO 69
69 COLOR 14
LOCATE 18, 41: PRINT "      1.2          0.0       "

   END IF
68 'CONTINUE

'SIGNAL RANGE CHECK
IF (A(2) < .05 OR A(2) > 9.45) THEN
   FLAG16(2) = 1
   GOTO 70
   ELSE
   GOTO 71
   END IF
70 COLOR 12
LOCATE 4, 45: PRINT "SRT SEN'R OUT RGE"
   PS = 1
   GOTO 557
71 COLOR 14
IF (PS = 1) THEN
   GOTO 558
   ELSE
   GOTO 557
   END IF
```

```
558 LOCATE 4, 45: PRINT "              "
   PS = 0
LOCATE 4, 45: PRINT "+19.0       -11.0"
557 'CONTINUE

IF (C(1) < .1 OR C(1) > 10!) THEN
   FLAG16(2) = 1
   GOTO 72
   ELSE
   GOTO 73
   END IF
72 COLOR 12
LOCATE 15, 45: PRINT "PSTAT SEN'R OUT RGE"
   PSS = 1
   GOTO 731
73 COLOR 14
IF (PSS = 1) THEN
   GOTO 732
   ELSE
   GOTO 731
   END IF
732 LOCATE 15, 45: PRINT "                  "
   PSS = 0
LOCATE 15, 45: PRINT "+57.9       +14.7  "
731 'CONTINUE

IF (C(2) < .35 OR C(2) > 4.7) THEN
   FLAG16(2) = 1
   GOTO 74
   ELSE
   GOTO 761
   END IF
74 COLOR 12
LOCATE 16, 44: PRINT "TOTAL PRESS SEN'R OUT RGE"
GOTO 75
761 COLOR 14
LOCATE 16, 44: PRINT "+135.0       +14.7       "
75 'CONTINUE

'SENSOR RANGE CODE CHECK FOR PSTAT
IF (C(3) < 3! OR C(3) > 4.5) THEN
   FLAG16(2) = 1
   GOTO 76
   ELSE GOTO 77
   END IF
76 COLOR 12
LOCATE 23, 42: PRINT "RANGE CODE (FOR PSTAT) OUT OF RGE"
LOCATE 23, 35: PRINT USING "###.##"; C(3)
GOTO 766
77 LOCATE 23, 35: PRINT "                                        "

766 'CONTINUE

'IF NO ERROR CONDITION, THE RESET FLAG16

   FLAG16(2) = 0

'TEST 10 SUBROUTINE---THIS ROUTINE CALCULATES THE DIFFERENCE BETWEEN
'BETA (MSW NS & FS) AND GAMA (REF NS & FS) AND COMPARES THE ANGLE
'WITH A LIMIT VALUE OF L100 (L100 = 15.0 DEG).  IF THIS ANGLE IS EXCEEDE
```

```
'THE FLAG IS SET TRUE.
'CALCULATE THE ANGULAR DIFFERENCE BETWEEN GAMATF AND BETATF---COMBINED TWO
'CALLS TO TEST 10 INTO ONE!
FLAG25(1) = 0
IF ((GAMATF - BETATF) >= L100) THEN
   FLAG25(1) = 1
   COLOR 12
   LOCATE 7, 46: PRINT "MSW/REFTF ANG X'C'ED"
   GOTO 781
   ELSE
   GOTO 782
782 COLOR 14
LOCATE 7, 46: PRINT "+0.0        -4.5    "
END IF
781 'CONTINUE

IF (GAMATF <= .1) THEN
   FLAG25(1) = 1
   GOTO 80
   ELSE
   GOTO 801
   END IF
80 COLOR 12
LOCATE 11, 45: PRINT "REFTF IN SLP'REAM"
GTF = 1
GOTO 81
801 COLOR 14
   IF (GTF = 1) THEN
   GOTO 802
   ELSE
   GOTO 81
   END IF
802 LOCATE 11, 45: PRINT "                    "
   GTF = 0
   LOCATE 11, 45: PRINT "+15.0          0.0"
81 'CONTINUE

IF ((GAMATN - BETATN) >= L100) THEN
   FLAG25(1) = 1
   COLOR 12
   LOCATE 8, 46: PRINT "MSW/REFTN ANG X'C'ED"
   GOTO 82
   ELSE
   GOTO 83
83 COLOR 14
LOCATE 8, 46: PRINT "+0.0        -4.5    "
   END IF
82 'CONTINUE

IF (GAMATN <= .1) THEN
   FLAG25(1) = 1
   GOTO 84
   ELSE
   GOTO 851
   END IF
84 COLOR 12
LOCATE 12, 45: PRINT "REFTN IN SLP'REAM"
GTN = 1
GOTO 85
851 COLOR 14
```

```
IF (GTN = 1) THEN
   GOTO 852
   ELSE
   GOTO 85
   END IF
852 LOCATE 12, 45: PRINT "                              "
   GTN = 0
   LOCATE 12, 45: PRINT "+15.0          0.0"
85 'CONTINUE

'SECOND CALL TO TEST 10
FLAG25(2) = 0
IF ((GAMABF - BETABF) >= L100) THEN
   FLAG25(2) = 1
   COLOR 12
   LOCATE 9, 46: PRINT "MSW/REFBF ANG X'C'ED"
   GOTO 86
   ELSE
   GOTO 87
87 COLOR 14
LOCATE 9, 46: PRINT "+0.0          -4.5    "
END IF
86 'CONTINUE

IF (GAMABF <= .1) THEN
   FLAG25(2) = 1
   GOTO 88
   ELSE
   GOTO 891
   END IF
88 COLOR 12
LOCATE 13, 45: PRINT "REFBF IN SLP'REAM"
   GBF = 1
   GOTO 89
891 COLOR 14
IF (GBF = 1) THEN
   GOTO 892
   ELSE
   GOTO 89
   END IF
892 LOCATE 13, 45: PRINT "                              "
   GBF = 0
   LOCATE 13, 45: PRINT "+15.0          0.0"
89 'CONTINUE

IF ((GAMABN - BETABN) >= L100) THEN
   FLAG25(2) = 1
   COLOR 12
   LOCATE 10, 46: PRINT "MSW/REFBN ANG X'C'ED"
   GOTO 90
   ELSE
   GOTO 91
91 COLOR 14
LOCATE 10, 46: PRINT "+0.0          -4.5    "
END IF
90 'CONTINUE

IF (GAMABN <= .1) THEN
   FLAG25(2) = 1
   GOTO 92
```

```
    ELSE
    GOTO 931
    END IF
 92 COLOR 12
 LOCATE 14, 45: PRINT "REFBN IN SLP'REAM"
    GBN = 1
    GOTO 93
 931 COLOR 14
 IF (GBN = 1) THEN
    GOTO 932
    ELSE
    GOTO 93
    END IF
 932 LOCATE 14, 45: PRINT "                    "
    GBN = 0
    LOCATE 14, 45: PRINT "+15.0         0.0"
 93 'CONTINUE

 'TEST 11 SUBROUTINE--THIS TESTS THE MEASURED MODEL ROLL ANGLE TO INSURE
 'THAT THE ROLL ANGLE DOES NOT EXCEED +275 OR -95.0 DEGREES.  IF THIS IS
 'EXCEEDED, A FLAG IS SET FOR OUTPUT TO THE SEQUENCER INTERLOCK.

 'COMPARE THE ROLL ANGLE WITH ITS PLUS AND MINUS LIMITS, AND SET A FLAG
 'TRUE IF LIMITS ARE EXCEEDED. L110A = +275 DEG , L110 = -95 DEG.
 FLAG25(4) = 0
 IF (ROLPOT >= L110A) THEN
    FLAG25(4) = 1
 ELSEIF (ROLPOT <= L110) THEN
    FLAG25(4) = 1
 END IF
 IF (FLAG25(4) = 1) THEN
    COLOR 12
    LOCATE 17, 44: PRINT "ROLL ANG X'C'ED"
    GOTO 96
    ELSE
    GOTO 97
 97 COLOR 14
    LOCATE 17, 44: PRINT "+275.0       -95.0"
    END IF
 96 'CONTINUE
 JJJ = JJ + 1
 JJ = JJJ
 'LOCATE 22, 55: PRINT USING "#####"; JJ

 'DEFINE THE DIGITAL I/O CONSTANTS---SEE "DT-2817 USER MANUAL"
 'CONTROL REGISTER WITH FACTORY BASE ADDRESS
 CONTROL.REGISTER% = &H250
 'OUTPUT PORT ADDRESSES
 DATA.PORT6 = &H251
 DATA.PORT7 = &H252
 DATA.PORT8 = &H253
 DATA.PORT9 = &H254
 'ALL PORTS SET FOR OUTPUT (&HF=1111)
 OUT CONTROL.REGISTER%, &HF
 DATA.VALUE(6) = TQ
 DATA.VALUE(7) = STRTMW
 DATA.VALUE(8) = TMR
 DATA.VALUE(9) = BMSWS
 'OUTPUT TO THE PORTS
 OUT DATA.PORT6, DATA.VALUE(6)
```

```
OUT DATA.PORT7, DATA.VALUE(7)
OUT DATA.PORT8, DATA.VALUE(8)
OUT DATA.PORT9, DATA.VALUE(9)
'READ BACK DATA THAT WAS OUTPUT ON THE PARTICULAR PORT
PORT.VALUE.READ(6) = INP(DATA.PORT6)
   XYZ = PORT.VALUE.READ(6)
PORT.VALUE.READ(7) = INP(DATA.PORT7)
   WXYZ = PORT.VALUE.READ(7)
PORT.VALUE.READ(8) = INP(DATA.PORT8)
   VWXYZ = PORT.VALUE.READ(8)
PORT.VALUE.READ(9) = INP(DATA.PORT9)
   UVWXYZ = PORT.VALUE.READ(9)
LOCATE 21, 54: PRINT USING "##"; XYZ
LOCATE 21, 56: PRINT USING "##"; WXYZ
LOCATE 21, 60: PRINT USING "##"; VWXYZ
LOCATE 21, 58: PRINT USING "##"; UVWXYZ

GOTO 1
```

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** July 1993	**3. REPORT TYPE AND DATES COVERED** Technical Memorandum

4. TITLE AND SUBTITLE A PC-based Simulation of the National Transonic Facility's Safety Microprocessor	**5. FUNDING NUMBERS** 505-59-85-01
6. AUTHOR(S) J. J. Thibodeaux, W. A. Kilgore, and S. Balakrishna	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-0001	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER** NASA TM-109003

11. SUPPLEMENTARY NOTES
J.J. Thibodeaux, NASA Langley Research Center, Hampton, VA
W.A. Kilgore, ViGYAN, Inc., Hampton, VA
S. Balakrishna, ViGYAN, Inc., Hampton, VA

12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category: 62	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(Maximum 200 words)*

A brief study was undertaken to demonstrate the feasibility of using a state-of-the-art off-the-shelf high speed personal computer for simulating a microprocessor presently used for windtunnel safety purposes at Langley Research Center's National Transonic Facility (NTF). Currently, there is no active display of tunnel alarm/alert safety information provided to the tunnel operators, but rather such information is periodically recorded on a process monitoring computer printout. This does not provide on-line situational information nor permit rapid identification of safety operational violations which are able to halt tunnel operations. It was therefore decided to simulate the existing algorithms and briefly evaluate a real-time display which could provide both position and trouble shooting information.

14. SUBJECT TERMS Control Systems, Personal Computers, Safety Systems	**15. NUMBER OF PAGES** 76
	16. PRICE CODE A05

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT**